

Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины»

А. С. ПОБИЯХА

**ОСНОВЫ АЛГОРИТМИЗАЦИИ
И ПРОГРАММИРОВАНИЯ**

Тестовые задания
для подготовки к экзамену
по курсу «Основы алгоритмизации и программирования»
для студентов специальности
1-39 03 02 Программируемые мобильные системы

Гомель
ГГУ им. Ф. Скорины
2022

УДК 004.42(076)
ББК 32.973я73
П417

Рецензенты:

кандидат физико-математических наук В. А. Савастенко,
кандидат технических наук А. С. Руденков

Рекомендовано к изданию научно-методическим советом
учреждения образования «Гомельский государственный
университет имени Франциска Скорины»

Побияха, А. С.

П417 Основы алгоритмизации и программирования : тестовые задания / А. С. Побияха ; Гомельский гос. ун-т им. Ф. Скорины. – Гомель : ГГУ им. Ф. Скорины, 2022. – 37 с.
ISBN 978-985-577-820-3

В настоящем пособии содержатся тестовые задания для самоконтроля и закрепления материала, пройденного в рамках лекций, а также лабораторных занятий по курсу «Основы алгоритмизации и программирования» для студентов специальности 11-39 03 02 «Программируемые мобильные системы».

УДК 004.42(076)
ББК 32.973я73

ISBN 978-985-577-820-3

© Побияха А. С., 2022
© Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины», 2022

ОГЛАВЛЕНИЕ

Предисловие.....	4
Тема 1. Операции и выражения.....	5
Тема 2. Ветвление.....	8
Тема 3. Циклы.....	13
Тема 4. Указатели.....	17
Тема 5 . Массивы.....	22
Тема 6. Функции пользователя.....	26
Тема 7. Типы данных.....	30
Тема 8. Файлы. Структуры, объединения, перечисления.....	34
Литература.....	37

ПРЕДИСЛОВИЕ

Курс «Основы алгоритмизации и программирования» для специальности 11-39 03 02 «Программируемые мобильные системы» является базовым, поскольку предполагает изучение основ программирования на языке Си, а также освоение базовых алгоритмов для работы с данными. Знания, полученные в рамках изучения материалов курса, могут быть использованы при освоении других языков программирования и технологий работы с данными, что наделяет данный курс статусом базового. Поэтому важно в процессе изучения студентами курса «Основы алгоритмизации и программирования» производить систематический самоконтроль с использованием тестовых заданий.

В издании содержатся тестовые задания для самоконтроля по темам курса «Основы алгоритмизации и программирования». Каждая тема содержит некоторое количество заданий для закрепления материала, пройденного в рамках лекций и лабораторных занятий.

Важно понимать, что самопроверка с использованием тестовых заданий связана с необходимостью предварительного изучения определенной теоретической части курса.

Поскольку издание является необходимой, но не достаточной частью процесса обучения, крайне важно при работе с данными тестовыми заданиями ознакомиться с соответствующими темами из таких источников информации, как: конспект лекций, книги, интернет. Таким образом, используя для подготовки к самоконтролю и экзамену несколько источников информации, можно получить качественную и точную картину изучаемой темы.

Самоконтроль выступает одним из важнейших компонентов обучения, поскольку позволяет развить у студентов умения самостоятельно управлять учебной деятельностью, контролировать и оценивать ее результаты, что, в свою очередь, позволит студенту эффективно использовать процесс овладения знаниями.

Тестовые задания написаны автором на основе многолетнего опыта проведения занятий по курсу «Основы алгоритмизации и программирования».

ТЕМА 1. ОПЕРАЦИИ И ВЫРАЖЕНИЯ

1. В каких выражениях используются бинарные арифметические операции?

- а) $X + Y$;
- б) $c \% d + 2$;
- в) $xx * Y$;
- г) $xx++$;
- д) $--xx$.

2. Для изменения порядка выполнения операций используется(-ются)...

- а) квадратные скобки;
- б) фигурные скобки;
- в) операторные скобки;
- г) знак косой черты;
- д) круглые скобки.

3. Выражение вида $x /= y + 15$; эквивалентно записи:

- а) $x = x / (y + 15)$;
- б) $x = y(x + 15)$;
- в) $y = y(x + 15)$;
- г) $y = x(y + 15)$;
- д) $x = y(y + 15)$;

4. Для какой арифметической операции используется знак «%»?

- а) умножение;
- б) вычисление процента;
- в) целая часть от деления;
- г) остаток от деления;
- д) инкремент.

5. Как обозначается логическая операция «И» в языке Си?

- а) $\&\&$;
- б) And ;
- в) $\&$;
- г) $!\&$;
- д) \parallel .

6. Каков будет результат выражения $!(1 \ \&\& \ !(0 \ || \ 1))$?

- а) true;
- б) false;
- в) undefined;
- г) ошибка синтаксиса;
- д) зависит от решаемой задачи.

7. Какой из следующих операторов – оператор сравнения двух переменных в Си?

- а) оператор « $=$ »;
- б) оператор « $+=$ »;
- в) оператор « $=$ »;
- г) оператор « $:=$ »;
- д) оператор « equ ».

8. Какой будет результат вычисления следующего выражения:
 $0xFF \ \& \ 5 \ \gg \ 1 + 1$; ?

- а) 1;
- б) 2;
- в) 5;
- г) 255;
- д) в Си нельзя выполнять такие операции.

9. Какой смысл имеет запись вида: $k = ++i$?

- а) $i = k$; $k = k + 1$;
- б) $i = i + 1$; $k = i$;
- в) $k = i$; $i = i + 1$;
- г) $k = i + 1$; $k = i$;
- д) $k = k + 1$; $i = k$;

10. Какой смысл имеет запись вида: $k = i++$?

- а) $i = i + 1$; $k = i$;
- б) $k = i + 1$;
- в) $k = i$; $i = i + 1$;
- г) $k = k + 1$; $i = k$;
- д) $I = k$; $k = k + 1$;

11. Нелогической операцией является...

- а) $\&\&$;
- б) $=$;

- в) ||;
- г) !;
- д) ==.

12. В результате выполнения следующего фрагмента кода `!(1 || 0) && 0` получается...

- а) 0;
- б) ошибка компиляции;
- в) 1;
- г) что результат определяется программистом;
- д) что результат не может быть заранее определён.

13. Какой будет результат выполнения операции `11 << 11`...

- а) 22;
- б) 33;
- в) 34;
- г) 22528.

14. Какой будет результат выполнения операции `34 >> 2`...

- а) 7;
- б) 8;
- в) 16;
- г) 15.

15. Укажите группу, содержащую последовательность правильно записанных на языке Си знаков операций отношений:

- а) `~ ~ > , < , = , ?`;
- б) `~ = , < > , > < >`;
- в) `~ ~ = > , = < , = , <`;
- г) `== , > = , < = , !=`;
- д) `~ > < , & , | , | >`.

ТЕМА 2. ВЕТВЛЕНИЕ

1. Если `int n = 3`, какой будет результат?

```
switch( n ) {  
    case 2: printf("aaa"); break;  
    case 3: printf("bbb"); break;  
    default: printf("vvv"); break;  
}
```

- а) aaa;
- б) bbb;
- в) vvv;
- г) aaabbb;
- д) bbbvvv.

2. Если `int n = 45`, какой будет результат?

```
switch( n ) {  
    case 23: printf("aaa");  
    case 45: printf("bbb");  
    default: printf("vvv");  
}
```

- а) aaa;
- б) bbb;
- в) vvv;
- г) aaabbb;
- д) bbbvvv.

3. Если условие оператора выбора ложное, то...

- а) выполняется следующий после `if` оператор;
- б) программа завершает работу;
- в) выполняется тело оператора выбора;
- г) результат работы программы неопределён;
- д) компилятор выдаёт ошибку.

4. Какое значение будет напечатано при выполнении фрагмента кода?

```
int x = 0, y = 0;  
if (x++ && y++){  
    y += 2;  
}  
printf("%d",x+y);
```


- а) 0;
- б) 1;
- в) 2;
- г) 3;
- д) 4.

5. Какой из ниже перечисленных вариантов ответа показывает правильно записанный оператор выбора?

- а) условное выражение if;
- б) if \{ условное выражение\};
- в) if (условное выражение);
- г) if условное выражение;
- д) else условное выражение if.

6. Какой служебный знак ставится после оператора case?

- а) * (звездочка);
- б) ; (точка с запятой);
- в) - (минус);
- г) : (двоеточие);
- д) . (точка).

7. Какому зарезервированному слову программа передаёт управление в случае, если значение переменной или выражения оператора switch не совпадает ни с одним константным выражением?

- а) next;
- б) previous;
- в) other;
- г) contingency;
- д) default.

8. Что будет напечатано при выполнении фрагмента кода?

```
int x = 3, y = 2, z = 1;
if (x >= y)
    if (y = z)
        printf("Вариант 2");
    else
        printf("Вариант 3");
```

- а) Вариант 2;
- б) Вариант 3;
- в) ничего не будет напечатано;

- г) произойдет ошибка компиляции в третьей строке;
- д) в Си нельзя выполнять такие операции.

9. В результате работы программы...

```
int main(){
    int ic;
    printf( "Введите любую десятичную цифру:");
    scanf("%d",&ic);
    switch (ic){
        case 0: case 1: printf("Один ");
        case 2: case 3: printf("Три ");
        case 4: case 5: printf("Пять ");
        case 6: case 7: printf("Семь ");
        case 8: case 9: printf("Девять "); break;
        default: printf("ERROR!!!");
    }
    return 1;
}
```

- а) в любом случае выводится "ERROR!!!";
- б) если введена нечетная цифра, выводится ее название;
- в) выводятся названия всех нечетных цифр больше или равных введенной;
- г) компилятор найдет ошибку;
- д) программа запустится, но «зависнет».

10. Результатом выполнения участка кода будет...

```
int i = 1, n = 5, k = 10;
if (i > 0 && i < n) k++;
```

- а) k = 10;
- б) k = 11;
- в) k = 9; i = 2;
- г) k = 9; n = 1;
- д) k = 10; n = 5;

11. Что будет выведено на экран при i равном 10?

```
switch(i){
    case 0: printf("Переменная равна нулю\n"); break;
    case 1: printf("Переменная равна единице\n"); break;
    default: printf("Переменная равна %d\n", i++); break;
}
```

- а) Переменная равна 10;
- б) Переменная равна единице;
- в) Переменная равна нулю;
- г) Ничего не выведет;
- д) Переменная равна 11.

12. Что будет выведено на экран при i равном 4?

```
switch(i){  
    case 0: printf("Переменная равна нулю\n"); break;  
    case 1: printf("Переменная равна единице\n"); break;  
    case 4: printf("Переменная равна %d\n", ++i); break;  
}
```

- а) Переменная равна единице;
- б) Переменная равна 5;
- в) Переменная равна нулю;
- г) Ничего не выведет;
- д) Переменная равна 4.

13. Как будет выполняться программа, если в переключателе switch не использовать оператор break?

- а) при совпадении ключевого выражения с меткой выполняется соответствующий оператор и управление передаётся на следующую ветвь case;
- б) при совпадении ключевого выражения с меткой выполняется соответствующий оператор и управление передаётся в конец переключателя;
- в) компилятор обнаружит синтаксическую ошибку;
- г) при совпадении ключевого выражения с меткой выполняется соответствующий оператор и управление передаётся на ветвь default;
- д) программа войдет в бесконечный цикл.

14. Какое значение будет находиться в переменной y после выполнения участка кода?

```
int y = 0 , x = 0;  
if (x = 0){y += 10;}
```

- а) 10;
- б) 11;
- в) 0;
- г) -10;
- д) будет ошибка компиляции.

15. Какие из операторов допустимо использовать внутри условия if?

- а) набор операторов «~, <, =, ?»;
- б) набор операторов «:, <>, ><, >»;
- в) набор операторов «|, =<, >=, !=»;
- г) набор операторов «==, >=, <=, !=»;
- д) набор операторов «><, √, /, |>».

РЕПОЗИТОРИЙ ГГУ ИМЕНИ Ф. СКОРИНЫ

ТЕМА 3. ЦИКЛЫ

1. В каком выражении произойдет заикливание программы?

- а) `for (int iCount = 0; iCount <= 4; iCount++);`
- б) `while (false);`
- в) `while (true);`
- г) `for (int iCount = 10; iCount >= 4; iCount--);`
- д) `while (true) break;`

2. До каких пор будут выполняться операторы в теле цикла `while (x < 100)?`

- а) пока `x` больше ста;
- б) пока `x` строго меньше ста;
- в) пока `x` делится на сто;
- г) пока `x` равен ста;
- д) пока `x` меньше или равен ста.

3. Как будет работать нижеприведенный фрагмент кода?

```
char ch ;  
for(ch = 0; ch < 256; )  
printf("%c-%d ", ch, ch++);
```

- а) фрагмент будет непрерывно выводить на экран половину таблицы ASCII;
- б) фрагмент будет непрерывно выводить на экран всю таблицу ASCII;
- в) фрагмент не может быть откомпилирован из-за ошибки в операторе `for`;
- г) фрагмент выведет на экран половину таблицы ASCII и закончит работу;
- д) фрагмент выведет на экран всю таблицу ASCII и закончит работу.

4. Какая из записей оператора `for` является правильной при вычислении $n!$?

- а) `for (i=1;i<=n;i=i+1) f=f*i;`
- б) `for (i=1,f=1;i<=n;n++) f=f*i+1;`
- в) `for (i=1,f=1;i<=n;n++) f=f*i;`
- г) `for (f=1,i=1;i<=n;i++) f*=i;`
- д) `for (f=1;i=1;i<=n;i++) f*=i;`

5. Какие значения будут выведены в консоль?

```
int i=5;
do{
    printf (" %d", ++i);
} while (i>=5 && i<=8);
```

- а) 6 7 8;
- б) 5 6 7;
- в) 6 7 8 9;
- г) ошибка компиляции;
- д) 5 6 7 8.

6. Какие из следующих конструкций корректны?

а) unsigned int i = 0;
do {
i--;
} while (i < 10);

б) int i = 0;
while (i < 10) {
i++;
}

в) int i = 0;
do {
i++;
} do (i < 10);

г) int i = 0;
while(i < 10) {
i++;
} while (i < 10);

д) int i = 0;
do(i < 10) {
i++;
}

7. Какие последние значения **x** и **y** будут отпечатаны оператором?

```
for ( x = 0, y = 1000; y > 1; x++, y/= 10)
    printf ( "%d%d\n", x, y);
```

- а) 1 и 100;
- б) 2 и 10;
- в) 3 и 1;
- г) 4 и 0;
- д) 0 и 1000.

8. Какими операторами можно организовать бесконечный цикл?

- а) `while(0) ;`
- б) `for(i = 0; 1; ++i) ;`
- в) `while(1) ;`
- г) `for(i = 0; TRUE; ++i) ;`
- д) `for(;;)`.

9. Какие из ниже перечисленных операторов не являются циклом в Си?

- а) `for`;
- б) `do while`;
- в) `repeat until`;
- г) `while`;
- д) `for in`.

10. Какой цикл всегда выполнится хотя бы один раз?

- а) цикл с не заданным заранее количеством повторений `while {...}`;
- б) цикл `for` с любым параметром;
- в) цикл с не заданным заранее количеством повторений `do {...} while;`
- г) цикл `for` с параметром равным нулю;
- д) цикл `for` в цикле `for`.

11. Корректен ли следующий фрагмент кода?

```
for(;;) {  
    // некоторый код  
}
```

- а) нет;
- б) да, если использовать соответствующие директивы препроцессора;
- в) да;
- г) да, если использовать его в отдельной функции;
- д) да, если запускать программу из консоли.

12. Правильно ли написан данный код?

```
int main()  
{  
    int x=1,i;  
    M2::  
    for(i=1;i<5;i++,x++);  
}
```

```
    if(x>6)
        goto M1;
    else
        goto M2;
M1: printf("%d",x);
    getch();
    return 0;
}
```

- а) нет, не правильно, ошибка в четвертой строке;
- б) да, правильно. Выведет число 9;
- в) нет, не правильно, ошибка в десятой строке;
- г) нет, не правильно, ошибка в пятой строке.

13. Сколько раз выполнится цикл?

```
int i;
while(i < 10){
    printf("%d-ый раз\n", i+1);
    i++;
}
```

- а) 10;
- б) 9;
- в) 11;
- г) произойдет ошибка компиляции;
- д) программа войдет в бесконечный цикл.

14. Цикл с постусловием – это...

- а) switch;
- б) for;
- в) if else;
- г) do while;
- д) while.

15. Чему будет равна переменная **a**, после выполнения этого кода?

```
int a;
for(a = 0; a <= 10; a++) { }
```

- а) переменная не изменится;
- б) 10;
- в) 9;
- г) 0;
- д) 11.

ТЕМА 4. УКАЗАТЕЛИ

1. В какой из следующих записей используется операция взятия адреса?

- а) a;
- б) address(a);
- в) addr(a);
- г) *a;
- д) &a;

2. В какой из следующих записей используется операция разыменования?

- а) &a;
- б) address(a);
- в) addr(a);
- г) *a;
- д) a;

3. В каком из фрагментов программы будет выведено *k=10; *i=14;?

- а)

```
int *k=new int,*i;  
*k=10;  
i=new int;*i=14;  
printf(“*k=%d *i=%d”,*k,*i);
```
- б)

```
int *k=new int,*i=new int;  
*k=10; i=k;  
*i=14;  
printf(“*k=%d *i=%d”,*k,*i);
```
- в)

```
int *k=new int,*i=new int;  
k=10; i=14;  
printf(“*k=%d *i=%d”,k,i);
```
- г)

```
int *k=new int,*i=new int;  
*k=10; *i=14;  
i=k;  
printf(“*k=%d *i=%d”,*k,*i);
```

4. Для получения адреса переменной используется операция...

- а) ->;
- б) . ;
- в) *;

- г) &;
- д) !=.

5. Как объявить переменную **a**, являющуюся указателем на константный указатель типа **T**?

- а) * const *T a;
- б) const *T*a;
- в) const T **a;
- г) этого сделать нельзя;
- д) T * const * a;

6. Какая из следующих записей возвращает значение переменной **a**, хранящееся в памяти по адресу, на который указывает указатель?

- а) value(a);
- б) *a;
- в) &a;
- г) val(a);
- д) a;

7. Какие операции допустимы для указателей **p1** и **p2**, объявленных как `int *p1, *p2;`?

- а) `p1 + p2;`
- б) `p1 - p2;`
- в) `*p1 + p2;`
- г) `*p1 **p2;`
- д) `p1 / p2.`

8. Каким кодом можно заменить постфиксный инкремент в строке с комментарием 1?

```
main() {  
    int a = 1;  
    int b = 2;  
    int * array[] = { &a, &b };  
    int ** pp = array;  
    (*++pp)++; // 1  
}
```

- а) `array[0]+=1;`
- б) `a+=1;`

- в) `array[1]+=1;`
- г) `b+=1;`
- д) `pp+=1;`

9. Какой тип указателя может указывать на любой тип данных?

- а) `const void ptr;`
- б) `void (*ptr)();`
- в) `void ptr;`
- г) такого указателя не существует;
- д) `void *ptr;`

10. Объявление `char *buf;` соответствует...

- а) созданию символьной переменной `buf;`
- б) созданию строковой переменной `buf;`
- в) созданию указателя `buf` на строку;
- г) созданию указателя `buf` на символьное значение;
- д) созданию указателя на массив символов `buf.`

11. Пусть дан массив `int a[6];`. Какой из способов доступа к 3-му элементу не является правильным?

- а) `int x, a[6]={1,2,3,4,5,6};`
`x=&a[0]+2;`
`printf("x=%d",x);`
- б) `int x, a[6]={1,2,3,4,5,6};`
`x=(a+2);`
`printf("x=%d",x);`
- в) `int x,*p, a[6]={1,2,3,4,5,6};`
`*p=(a+2);`
`printf("x=%d",*p);`
- г) `int x,*p, a[6]={1,2,3,4,5,6};`
`p=a+2;`
`printf("x=%d",*p);`

12. Перед какими операторами `printf` надо убрать символ комментария, чтобы было напечатано максимальное количество 200?

```
int data[3][7]={{1,2,3,4,5,6,7},{10,20,30,40,50,60,70},
{100,200,300,400,500,600,700}};
int *pi=&data[1][1], (*pd)[7]=data;
// printf(" %d ",data[2][1]);
```

```
// printf(" %d ",data[3][2]);  
// printf(" %d ",*(*(pd+2)+1));  
// printf(" %d ",*((*pd+2)+1));  
// printf(" %d ",*(*(pd+1)+2));  
// printf(" %d ",*(pi+7));  
// printf(" %d ",*(pi+7));
```

- а) 1;
- б) 2;
- в) 3;
- г) 4;
- д) 5;
- е) 6;
- ж) 7.

13. Укажите правильное объявление указателя в Си:

- а) int *x;
- б) ptr x;
- в) pointer x;
- г) int x;
- д) int &x;

14. Что напечатает этот фрагмент кода?

```
int z, a[]={2,3}, *p=a;  
z=++*++p;  
printf("%5d%5d",z,*p);
```

- а) 3 3;
- б) 4 2;
- в) 4 3;
- г) 4 4;
- д) 4 1.

15. Что произойдет в результате компиляции и выполнения следующего кода?

```
int main() {  
    int a = 5;  
    int* b = &a; // 1  
    printf("%d", a**b); // 2
```

```
    return 0;  
}
```

- а) возникнет ошибка времени выполнения;
- б) возникнет ошибка компиляции в строке 1;
- в) возникнет ошибка компиляции в строке 2;
- г) будет напечатано число 25;
- д) будет напечатано некоторое целое число (зависит от адреса переменной b).

РЕПОЗИТОРИЙ ГГУ ИМЕНИ Ф. СКОРИНЫ

ТЕМА 5 . МАССИВЫ

1. Массив – это...

а) упорядоченные в памяти элементы одного и того же типа, имеющие общий адрес. Доступ к отдельным элементам массива осуществляется по адресу элемента и индексу;

б) упорядоченные в памяти элементы одного и того же типа, имеющие имя. Доступ к отдельным элементам массива осуществляется по имени массива и индексу;

в) упорядоченные в памяти элементы одного и того же типа, имеющие имя. Доступ к отдельным элементам массива осуществляется по имени массива и адресу.

2. В каком из вариантов ответов объявлен двумерный массив?

а) `int anarray[20][20];`

б) `char array[20];`

в) `int anarray(20,20);`

г) `int array[20, 20];`

д) `array anarray[20][20];`

3. Дан массив `int L[3][3] = { { 2, 3, 4 }, { 3, 4, 8 }, { 1, 0, 9 } }`; Чему будет равно значение элемента этого массива `L[1][2]`?

а) 9;

б) 4;

в) 8;

г) 2;

д) 3.

4. Какой массив имеет самый большой размер?

`char *s1= "QWERTY";`

`char s2[7]= "qwerty";`

`char *s3= "127*27";`

а) s1;

б) s3;

в) все массивы имеют одинаковый размер;

г) s2.

5. Какой порядковый номер последнего элемента массива при размере массива 19?

- а) i+1;
- б) 20;
- в) определяется программистом;
- г) 19;
- д) 18.

6. Какой размер массива M будет после выполнения кода:
`char M[]="\nGoodlive"?`

- а) 9;
- б) 11;
- в) 13;
- г) 8;
- д) 10.

7. Какой будет результат выполнения следующего участка кода?

```
int *a;  
int b[2];  
a = b;  
b[0] = 7;  
b[1] = 10;  
*a++;  
printf("%d",*a);
```

- а) 11;
- б) 10;
- в) 8;
- г) 7;
- д) ошибка компиляции в третьей строке.

8. Массив инициализирован следующим образом: `int x[5]={1,2,3,4,5};`
Какая ошибка будет при следующей записи оператора `x[4]=x[5]+1;` ?

- а) неверная инициализация;
- б) ошибка прав доступа;
- в) синтаксическая ошибка;
- г) нарушение границы массива;
- д) оператор записан верно.

9. Пользователь вводит строку "Иванов Иван Иванович". Эту строку программа читает после нажатия клавиши Enter с помощью функции `scanf("%s", str)`. Что будет занесено в массив `str`?

- а) "Иванов Иван Иванович";
- б) "Иванов";
- в) "Иван Иванович";
- г) ничего, так как неверно задан формат ввода строки;
- д) ничего, так как перед именем str не хватает символа &.

10. Пусть объявлена и инициализирована матрица:

```
float a[3][3]={ 1,-0.5,3,2,1.5,3,-2,3,1};
```

Требуется вывести матрицу по строкам.

Какой из участков кода делает это правильно?

- а)

```
for (i=0;i<=3;i++) { for (j=0;j<=3;j++) printf(“%6.2f ”,a[i][j]);
printf(“\n”); }
```
- б)

```
for (i=0;i<3;i++) for (j=0;j<3;j++) printf(“%6.2f ”,a[i][j]); printf(“\n”);
```
- в)

```
for (i=0;i<3;i++) { for (j=0;j<3;j++) printf(“%6.2f ”,a[i][j]);
printf(“\n”); }
```
- г)

```
for (i=1;i<=3;i++) { for (j=1;j<=3;j++) printf(“%6.2f ”,a[i][j]);
printf(“\n”); }
```

11. Словосочетание "Hello world!" может быть сохранено в символьном массиве размером n элементов. Укажите чему равно n?

- а) 10;
- б) 12;
- в) 13;
- г) 11;
- д) нельзя строку записать в символьный массив.

12. Укажите корректное определение строковой переменной.

- а) `string mystr;`
- б) `char mystr;`
- в) `string[20] mystr;`
- г) `string mystr[20];`
- д) `char mystr[20];`

13. В результате работы программы будет выведено следующее значение...

```
void main()
{
    short int a[] = {1,2,3};
    printf( "%d", sizeof(a)/sizeof(a[0]) );
}
```


- а) 6;
- б) 3;
- в) 12;
- г) 16;
- д) 4.

14. Что будет содержать массив **x** после выполнения кода, показанного ниже?

```
int x[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };  
int y = 20;  
int i = 10;  
x[--i] = y;
```

- а) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10;
- б) 1, 2, 3, 4, 5, 6, 7, 8, 9, 20;
- в) 1, 2, 3, 4, 5, 6, 7, 8, 20, 10;
- г) 20, 2, 3, 4, 5, 6, 7, 8, 9, 10.

15. Что будет напечатано в консоли в результате выполнения кода?

```
void main()  
{  
    char a[] = "123";  
    printf( "%d", strlen(a) );  
}
```

- а) 0;
- б) 1;
- в) 3;
- г) -1;
- д) 4.

ТЕМА 6. ФУНКЦИИ ПОЛЬЗОВАТЕЛЯ

1. В каких случаях необходимо использовать оператор return в теле функции?

- а) если необходимо обеспечить выход из функции в произвольном месте;
- б) если указан тип возвращаемого значения, в том числе и void;
- в) если необходимо, чтобы выполнение функции началось с начала;
- г) если необходимо, чтобы функция вернула значение;
- д) всегда.

2. Во что заключается тело функции?

- а) в одинарные кавычки;
- б) в квадратные скобки;
- в) в фигурные скобки;
- г) в двойные кавычки;
- д) в круглые скобки.

3. Как изменяются локальные и глобальные объекты в рекурсиях?

- а) глобальные объекты не сохраняют свои значения после окончания рекурсии;
- б) локальные объекты сохраняют свои значения после окончания рекурсии;
- в) при повторных вызовах рекурсивной функции создается новое множество локальных объектов.

4. Как можно передать результат (тип float) выполнения функции типа void?

- а) через имя функции;
- б) посредством вызова подпрограммы;
- в) через формальный параметр типа float;
- г) через формальный параметр типа указатель на float;
- д) через формальный параметр типа void.

5. Как называется функция, которая вызывает саму себя?

- а) подставляемой;
- б) геттером;
- в) конструктором;
- г) рекурсивной;
- д) деструктором.

6. Какое из следующих утверждений об операторе return является верным?

- а) оператор return должен стоять последним в теле функции;
- б) в теле функции должен присутствовать только один оператор return;
- в) оператор return завершает выполнение функции;
- г) return нельзя использовать в функции;
- д) в конце функции всегда нужно ставить return 0;

7. Какой оператор должен обязательно присутствовать в рекурсивной функции?

- а) оператор цикла;
- б) оператор return;
- в) оператор передачи управления;
- г) оператор проверки окончания рекурсии;
- д) оператор умножения.

8. Какой результат вернет функция strcmp("a","b");?

- а) b;
- б) 1;
- в) a;
- г) -1;
- д) 0.

9. Ключевое слово void обозначает, что функция...

- а) возвращает число с плавающей запятой;
- б) ничего не возвращает;
- в) возвращает целое число;
- г) является главной;
- д) является второстепенной.

10. Какая функция является рекурсивной?

- а) для которой результат передаётся через имя функции;
- б) для которой в теле функции имеется оператор вызова этой же функции;
- в) для которой в теле функции присутствует оператор return;
- г) для которой результат передаётся через формальный параметр-указатель.

11. Какие из предложенных имен функций допустимы?

- а) calculateIt;
- б) _This_Function_12_x_;
- в) fffffffAAAAAA;
- г) ax%u7;
- д) 3_pi.

12. Прототип функции задает...

- а) тип функции, включая количество и тип аргументов и тип результата;
- б) возможность выполнения этой функции из программ на других языках программирования;
- в) имя функции и минимальное количество параметров;
- г) возможность в будущем совершенствовать функцию;
- д) правила запуска функций.

13. Сколько функций может быть в программе Си?

- а) не больше 100;
- б) минимум одна;
- в) ни одной;
- г) не больше 255;
- д) ровно 1024.

14. Что будет напечатано в результате выполнения программы?

```
void main()
{
    char ch[] = "10.123";
    *(ch + 4) = 0;
    printf("%.3f", atof(ch + 1));
}
```

- а) 11.103;
- б) 1.103;
- в) 0.103;
- г) 0.100.

15. Каков будет результат при запуске следующего кода?

```
int dbl(int); // 1
int main() {
    printf("%d", dbl(5)); // 2
    return 0;
}
```

- ```
 }
 int dbl(int a) { // 3
 return a * 2;
 }
а) 0;
б) 10;
в) ошибка в строке 1;
г) ошибка в строке 3;
д) ошибка в строке 2.
```

РЕПОЗИТОРИЙ ГГУ ИМЕНИ Ф. СКОРИНЫ

## ТЕМА 7. ТИПЫ ДАННЫХ

1. Если в арифметическом выражении участвуют короткое целое и длинное целое, то...

- а) короткое приводится к длинному;
- б) произойдет ошибка компиляции;
- в) короткое целое приводится к вещественному типу;
- г) длинное приводится к короткому;
- д) оба числа приводятся к вещественному типу.

2. Если в арифметическом выражении участвуют целый и вещественный операнды, то...

- а) вещественный тип приводится к целому;
- б) точность целого числа уменьшится;
- в) целый тип приводится к вещественному;
- г) точность вещественного числа уменьшится;
- д) произойдет ошибка компиляции.

3. Если есть два объявления `int qwerty; int QWERTY;` какое из утверждений верно?

- а) такие объявления правильные;
- б) такие имена переменных недопустимы;
- в) такие объявления недопустимы, так как мы пытаемся создать две одинаковые переменные;
- г) произойдет ошибка компиляции.

4. Как обозначается вещественный тип в языке Си?

- а) `int`;
- б) `void`;
- в) `float`;
- г) `real`;
- д) `char`.

5. Если размер типов данных в байтах `char=1`, `int=4` и `float=4`, то каким будет результат работы следующей программы?

```
int main(){
 char ch = 'A';
 printf("%d, %d, %d", sizeof(ch), sizeof('A'), sizeof(3.14f));
 return 0;
}
```

- а) 2, 2, 4;
- б) 2, 2, 2;
- в) 1, 1, 4;
- г) 1, 2, 4;
- д) 2, 4, 8.

6. Какая директива препроцессора позволяет создать константу?

- а) #const;
- б) #create;
- в) #include;
- г) #new;
- д) #define.

7. Какие из фундаментальных типов Си являются архитектурно-независимыми и имеют одинаковое представление на всех платформах?

- а) long;
- б) int;
- в) unsigned int;
- г) float;
- д) unsigned long;
- е) double;
- ж) никакие из перечисленных.

8. Какие преобразования типов данных невозможны без потери данных?

- а) int to float;
- б) float to int;
- в) char to float;
- г) все перечисленные преобразования невозможны;
- д) все перечисленные преобразования возможны.

9. Какой вид преобразования типов используется в следующем выражении?

```
int a = 0;
float f = 3.4;
a += f;
```

- а) явное преобразование типов;
- б) неявное преобразование типов;
- в) пользовательское преобразование типов;
- г) преобразование типов не происходит.

10. Какой диапазон значений имеет тип `unsigned int` для 32-разрядных вычислительных систем?

- а) от 0 до 65 535;
- б) от 0 до 1;
- в) от 0 до 255;
- г) от 0 до 4 294 967 295;
- д) от -32 768 до 32 767.

11. Какой из перечисленных типов данных не является типом данных в Си?

- а) `char`;
- б) `int`;
- в) `double`;
- г) `real`;
- д) `float`.

12. Какой размер в байтах имеет переменная вещественного типа `float`?

- а) 8;
- б) 6;
- в) 4;
- г) 1;
- д) 2.

13. Какой спецификатор предназначен для вещественного числа?

- а) `%d`;
- б) `%f`;
- в) `%x`;
- г) `%s`;
- д) `%o`.

14. Какой стандартный тип допустим в Си?

- а) `real`;
- б) `string`;
- в) `double`;
- г) `byte`;
- д) `longint`.



15. Сколько можно закодировать различных символов, используя тройку из нулей и единиц?

- а) 5;
- б) 6;
- в) 8;
- г) 9;
- д) 16.

РЕПОЗИТОРИЙ ГГУ ИМЕНИ Ф. СКОРИНЫ

## ТЕМА 8. ФАЙЛЫ. СТРУКТУРЫ, ОБЪЕДИНЕНИЯ, ПЕРЕЧИСЛЕНИЯ

1. В каких случаях целесообразно использовать файл?

- а) для быстрого поиска по заданным критериям;
- б) для сортировки данных;
- в) для математических вычислений;
- г) для длительного хранения больших объёмов данных.

2. Как открыть бинарный файл для изменения в нём информации?

- а) `f=fopen("file1.dat","wb+")`;
- б) `f=fopen("file1.dat","ab")`;
- в) `f=fopen("file1.dat","r+")`;
- г) `f=fopen("file1.dat","rb+")`;
- д) `f=fopen("file1.dat","re+")`;

3. Как правильно объявить указатель на файл f?

- а) `int *f`;
- б) `F &f`;
- в) `File &f`;
- г) `FILE *f`;
- д) `F *f`;

4. Как считать из файла строку str, состоящую из слов, разделённых пробелами?

- а) `fscanf(f,"%s",str)`;
- б) `str=fgetc(f)`;
- в) `fgets(str,80,f)`;
- г) `fread(str,sizeof(str),1,f)`;

5. Какая функция предназначена для открытия файла?

- а) `fileOpen`;
- б) `fopen`;
- в) `open`;
- г) `openFile`;
- д) `opfl`.

6. Какое действие выполняется при открытии файла?

- а) устанавливается тип файла (текстовый или бинарный);
- б) устанавливается курсор в начало файла;

- в) физический файл связывается с логическим (файловой переменной);
- г) возвращается результат (ошибка);
- д) устанавливается вид (режим) использования файла.

7. Логическое имя (идентификатор файла) – это...

- а) имя структуры данных, состоящей из битовых полей;
- б) указатель на файл, т. е. на область памяти, где содержится вся необходимая информация о файле;
- в) идентификатор сложной структуры данных, состоящей из чередующихся нулей и единиц;
- г) последовательность символов, каждая строка которой заканчивается двумя кодами «перевод строки», «возврат каретки»: 13 и 10 (0xD и 0xA).

8. В состав каких типов данных не может входить структура как компонент?

- а) файл;
- б) массив;
- в) extended;
- г) структура;
- д) может во все перечисленные.

9. Как можно обратиться к полю Year объекта Student типа структуры Person?

- а) Person.Student;
- б) Student.Year;
- в) Person.Year;
- г) scanf(“%d”, &Person.Year).

10. Какого типа могут быть элементы (поля) структуры?

- а) полями структуры могут быть файлы;
- б) поля структуры должны быть одного и того же типа;
- в) полями структуры могут быть другие структуры;
- г) полями структуры не могут быть массивы.

11. Каким будет значение элемента перечисления THIRD?

```
enum digits {
 FIRST = 0, SECOND = 0, THIRD
};
```

- а) -1;
- б) -2;

- в) 1;
- г) 2;
- д) 0.

12. Какое ключевое слово языка Си используется для описания структурированного типа данных, все элементы которого в памяти начинаются с одного байта?

- а) array;
- б) union;
- в) enum;
- г) template;
- д) struct.

13. Каким будет размер структуры?

```
struct {
 char fio[30];
 unsigned char date:4;
 unsigned char code:4;
};
```

- а) 16 байт;
- б) 31 байт;
- в) 9 байт;
- г) 38 байт;
- д) 8 байт.

14. С какого ключевого слова начинается описание объединения?

- а) struct;
- б) enum;
- в) int;
- г) typedef;
- д) union.

15. Каким будет значение элемента перечисления HIGH?

```
enum levels {
 LOW = 10, MEDIUM = 100, HIGH
};
```

- а) 1000;
- б) 101;
- в) 110;
- г) 1010;
- д) 0.

## ЛИТЕРАТУРА

1. Матвиенко, Ю. А. Использование самоконтроля учебной деятельности студентов экономических специальностей в процессе психолого-педагогической подготовки с целью повышения качества усвоения знаний / Ю. А. Матвиенко // Молодой ученый. – 2013. – № 5 (52). – С. 744–748.
2. Керниган, Б. Язык программирования С / Б. Керниган, Д. М. Ритчи. – 2-е изд. – М. : Вильямс, 2016. – 288 с.
3. Керниган, Б. У. Язык программирования С / Б. У. Керниган, Д. М. Ритчи ; пер. с англ. В. Л. Бродовой. – М. : Вильямс, 2013. – 304 с.
4. Ашарина, И. В. Основы программирования на языках С и С++ : курс лекций для высших учебных заведений / И. В. Ашарина. – М. : Гор. линия–Телеком, 2012. – 208 с.
5. Дорогов, В. Г. Основы программирования на языке С : учебное пособие / В. Г. Дорогов, Е. Г. Дорогова ; под общ. ред. проф. Л. Г. Гагарина. – М. : ИД ФОРУМ, НИЦ ИНФРА-М, 2013. – 224 с.
6. Полубенцева, М. С/С++. Процедурное программирование / М. Полубенцева. – М. : БХВ-Петербург, 2014. – 448 с.
7. Гукин, Д. Программирование на С для чайников / Д. Гукин. – М. : Диалектика, 2019. – 384 с.
8. МакГрат, М. Программирование на С для начинающих / М. МакГрат. – М. : Эксмо, 2015. – 192 с.
9. Перри, Г. Программирование на С для начинающих / Г. Перри, Д. Миллер. – М. : Эксмо, 2015. – 368 с.
10. Сикорд, Роберт С. Безопасное программирование на С и С++ / Роберт С. Сикорд. – Москва : РГГУ, 2014. – 496 с.
11. Гриффитс, Д. Изучаем программирование на С / Д. Гриффитс. – М. : Эксмо, 2015. – 624 с.

Учебное издание

**Побияха** Александр Сергеевич

## **ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ**

Тестовые задания

Редактор А. А. Негодина  
Корректор В. В. Калугина

Подписано в печать 21.01.2022. Формат 60x84 1/16.

Бумага офсетная. Ризография.

Усл. печ. л. 2,33. Уч.-изд. л. 2,54.

Тираж 25 экз. Заказ 34.

Издатель и полиграфическое исполнение:  
учреждение образования  
«Гомельский государственный университет  
имени Франциска Скорины».

Свидетельство о государственной регистрации издателя,  
изготовителя, распространителя печатных изданий № 3/1452 от 17.04.2017.

Специальное разрешение (лицензия) № 02330 / 450 от 18.12.2013.

Ул. Советская, 104, 246028, г. Гомель.

РЕПОЗИТОРИЙ ГГУ ИМЕНИ Ф. СКОРИНЫ

РЕПОЗИТОРИЙ ГГУ ИМЕНИ Ф. СКОРИНЫ