

Контрольная работа №2
09.12.2016

№№ 1-15 оцениваются в 2 балла.

Выполняйте задания рядом с условиями.

1. Дан массив: `int a[] = {-2, -1, 1, 2, 3, 4, 5, -1, -2}`. Сколько места занимает данный массив? Описать цикл, который выводит все положительные числа (без использования условного оператора).

2. Что будет напечатано?

```
int a[] = {3, 4, 5, 6, 7, 8, 9}, i, j;  
for (i = 1; i <= 3; i++)  
    for (j = 0; j <= 2; j++)  
        printf("%d ", a[i*j]);
```

3. Как изменится массив `int a[] = {0, 1, 2, 3, 4}` после следующих команд (выполняются друг за другом):

```
a[0]++;  
a[a[0]]++;  
a[a[a[0]]]++;
```

4. В каких случаях возникает ошибка `segmentation fault (core dumped)` и что она означает?

5. Что будет выдано на экран? Чему равно n после выполнения кода?

```
fgets(a, 100, stdin);  
a[10] = '\0';  
for (n=0; a[n]; n++)  
    puts(a);
```

6. Что выполняет код?

```
fgets(a, 100, stdin);  
fgets(b, 100, stdin);  
for (n=0; a[n] && b[n] && a[n] == b[n]; n++);  
if (a[n] && b[n] && a[n] > b[n])  
    puts(a);  
else  
    puts(b);
```

7. Какие из функций `scanf("%s", a)`, `gets(a)`, `fgets(a, 100, stdin)` являются безопасными и почему? Чем отличаются функции `puts(a)`, `printf("%s", a)`?

8. Как расположены элементы в памяти?

```
int a[2][3] = {{1, 2, 3}, {4, 5, 6}};
char a[2][] = {'a', 'b', 'c'}, {'d', 'e', 'f'};
char a[2][] = {"abc", "def"};
```

9. Найти ошибки в коде и пояснить их (программа должна в матрицу *b* записать квадрат матрицы *a*):

```
int a[3][3] = {{1, 2, 3} {4, 5, 6} {7, 8, 9}}, b[3][3];
for (i = 1; i <= 3; i++)
    for (j = 1; j <= 3; j++)
        for (k = 1; k <= 3; k++)
            b[i][j] = a[i][k] * a[k][j];
```

10. Зачем нужны функции? Что такое локальные переменные? Что такое адрес возврата, зачем он нужен и где он хранится?

11. Чему равны значения переменной *a* после каждого оператора присваивания?

```
int f(int n){
    if (n % 2 == 1) return n + 1;
    else n++;
    return n / 2;
}
...
a = f(1);
a = f(a);
a = f(f(a)+a);
```

12. Пусть *n* — целое положительное число от 1 до 10^5 . Что вычисляет программа (с точки зрения теории чисел)?

```
int alpha(int n, int d){
    int ans = 0;
    for (; n % d == 0; n /= d) ans++;
    return ans;
}
int f(int n){
```

```

int d, ans = 1;
for (d = 2; d <= n; d++) ans *= alpha(n, d) + 1;
return ans;
}

```

13. Нарисовать дерево вызовов данной функции, при вызове $f(4)$:

```

void f(int n){
    if (n <= 0)
        return;
    f(n-1);
    f(n/2);
}

```

Какого минимального размера должен быть системный стек чтобы корректно произошел вызов $f(10)$?

14. Исправьте ошибки в код (если строка является палиндромом, то функция должна возвращать 1, иначе — 0)?

```

int ok(int n, char s[]){
    if (s[0] == s[n])
        return 1;
    else
        return 0;
    return ok(n-2, s+1);
}

```

15. Описать рекурсивную функцию `int popcount(int n)`, которая вычисляет количество единичных бит в двоичной записи числа n .

№№ 16-20 оцениваются в 4 балла.

Выполняйте задания на обратной стороне листов.

16. Дана строка из строчных буквы английского алфавита длиной не более 100 символов. Распечатать все буквы в алфавитном порядке, которые встретились в строке. На следующей строке вывести буквы, которые встретились реже всего

Ввод	antananarivo
Вывод	a i n o r t v i r t v

17. Дана строка s из строчных букв английского алфавита длиной не более 100 символов. Построить z -функцию для данной строки: $z[i]$ — наибольший общий префикс подстроки $s[i..n]$ и строки s . Вывести значения z -функции. Поиск максимального префикса оформить в виде отдельной функции.

Ввод	babababab
Вывод	8 0 2 0 0 3 0 1

18. Дано целое положительное n от 1 до 10. Далее матрица размером $n \times n$ из целых чисел от -100 до 100. Строку с максимальным элементом в матрице переставить местами со строкой, где расположен минимальный элемент матрицы (гарантируется, что максимум и минимум в матрице встречается только 1 раз).

Ввод	5 1 2 3 4 2 1 0 1 1 2 3 2 1 6 2 5
Вывод	1 2 3 4 1 6 2 5 1 2 3 2 2 1 0 1

19. Дано целое положительное число n от 1 до 100. Найти все различные дроби из интервала $(0; 1)$, у которых числитель и знаменатель не превышает n без повторов.

Ввод	4
Вывод	1/4 1/3 1/2 2/3 3/4

20. Дано целое число n от 1 до 100 — размер массива. Далее n различных целых чисел — массив. Отсортировать массив быстрой сортировкой `qsort`.

Ввод	5 2 1 5 3 4	5 5 4 3 2 1	5 1 2 3 4 5
Вывод	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5

- 1) выберем ведущий элемент $p = a[m]$, где $m = (\text{left} + \text{right}) / 2$;
- 2) заведем 2 указателя $l = \text{left}$; $r = \text{right}$;
- 3) двигаем указатели навстречу друг другу до тех пор, пока указатели не встретятся или пока не найдутся элементы $a[l] \geq p$ и $p \geq a[r]$;
- 4а) если элементы нашлись, то меняем найденные элементы между собой, сдвигаем каждый из указателей еще на один шаг и продолжаем поиск;
- 4б) иначе сортируем подмассивы $a[\text{left}..r]$ и $a[l..right]$.