

Контрольная работа №3
29.12.2016

№№ 1-10 оцениваются в 2 балла.
Выполняйте задания рядом с условиями.

1. Дано описание структуры СТУДЕНТ (имя и 3 оценки)

```
struct student{ char name[5]; int mark[3];};
```

Сколько байт в памяти занимает один экземпляр данной структуры?

С помощью какой функции можно узнать этот размер в коде программы на языке C?

Описать переменную A данной структуры описывающего имя Dima и оценками 3, 4 и 5.

2. Описать 3 варианта типа переменной v (и соответствующей структуры), у которых корректны обращения:

```
v[5].x,  
v.x[5],  
v[5].x[3].
```

3. Дан динамический целочисленный массив с указателем `double * a` размером n элементов.

Описать 2 команды, которые перевыделяет память и добавляют к массиву в конец одно число (со значением 2016.2017).

Описать вызов одной функции, которая удаляет вторую половину массива.

4. Чем отличаются типы `int [10][20]`, `(int *)[20]`, `int (*[20])`, `int **`?

Что использовать как аргумент функции, описывающий статическую матрицу?

Что использовать как аргумент функции, описывающий динамическую матрицу?

5. Вывести значение переменные после каждого действия. Найти и исправить ошибку.

```
int a = 5;  
int *b = &a;  
a = 100/*b;  
a ** b;  
a += b[0];
```

6. Что будет напечатано:

```

char a[] = "mechanics";
char b[] = "mathematics";
char *c = NULL;
puts(a+5);
strcpy(b+1, a+5);
puts(b);
c = strstr(a+3, b+strlen(b)-1);
puts(c);

```

7. В качестве аргумента командной строки передано: `./prog one two three four`. Какое количество памяти в байтах занимают все аргументы командной строки `int main(int argc, char ** argv)` (включая динамическую память, на которую они ссылаются). Рассмотреть 2 варианта: 32-битная ОС и 64-битная ОС.
8. В чем отличие динамически и статически выделенной памяти:
 Когда выделяется память?
 Где выделяется память?
 Когда удаляется память?
 Что эффективней для матриц в плане скорости работы и памяти для матриц порядка 1000?
9. Написать программу, которая печатает свой исходный код (предполагается, что исполняемый файл имеет такое же название, что и исходный код без суффикса `*.c`, длина имени файла с исходным кодом не превышает 10 символов и программа запущена из текущей директории). Использовать корректное форматирование.

```

#include <stdio.h>
#include <string.h>
int main(int argc, char **argv){
    char fname[11], ch;
    FILE * fd;

    -----
    -----
    fd = fopen(fname, "r");
    while ( (ch = fgetc(fd)) != EOF) {
        -----
        -----
    }
    fclose(fd);
    return 0;
}

```

10. Что такое файловые дескрипторы?
 Какие стандартные файловые дескрипторы есть у любой программы?
 Как запустить программу в терминале так, чтобы вывод записывался в начало файла `out.txt`, а ошибки записывались в конец файла `err.txt`.

№№ 11-15 оцениваются в 4 балла.

11. Дано время в формате hh:mm. Далее дано целое положительное число s . Каждые s минут до конца суток должен звонить будильник. Выведите все отсечки времени, в которые звонил будильник. Использовать структуру ВРЕМЯ и функцию добавления к данной отсечки времени данного количества минут.

Ввод	09:00 200
Вывод	09:00 12:20 15:40 19:00 22:20

12. Даны 3 комплексных числа в формате $x+yi$ или $x-yi$ (где x и y — целые неотрицательные числа от 0 до 9). Вывести коэффициенты приведенного ($a = 1$) кубического уравнения $ax^3 + bx^2 + cx + d = 0$ с данными корнями. Использовать структуру КОМПЛЕКСНОЕ ЧИСЛО и функции необходимых арифметических операций.

Ввод	1+0i -1+0i 0+0i	1+1i 1-1i 0+1i
Вывод	1+0i 0+0i -1+0i 0+0i	1+0i -2-1i 2+2i 0-2i

Примеры:

1, -1 и 0 являются корнями уравнения $x^3 - x = 0$.

$1 + i$, $1 - i$ и i являются корнями уравнения $x^3 + (-2 - i)x^2 + (2 + 2i)x - 2i = 0$.

13. Дано целое число n от 1 до 100000 как аргумент командной строки. Найти все избыточные числа (сумма положительных собственных делителей отличных от n больше самого числа) в диапазоне от 1 до n . Далее вывести их медиану (средний элемент, если общее количество нечетно, и полусумма двух соседних средних элементов, если количество четно). Использовать динамический массив для хранения чисел.

запуск	./prog 55
Вывод	12 18 20 24 30 36 40 42 48 54 33

запуск	./prog 50
Вывод	12 18 20 24 30 36 40 42 48 30

14. В файле `base.txt` записано одно целое положительное n . Далее n строк с описанием n -мерных векторов e_i по строкам. Построить матрицу Грама G для данной системы векторов ($g_{ij} = (e_i, e_j)$) и вывести результат в файл `gram.txt`. Использовать динамическое выделение матрицы и функцию нахождения скалярного произведения.

<code>base.txt</code>	3 1 2 3 2 3 1 3 1 0
<code>gram.txt</code>	14 11 5 11 14 9 5 9 10

15. В качестве аргумента командной строки дано имя файла и строка. Распечатать все строки из файла (первый аргумент), которые содержат подстроку (второй аргумент). Использовать динамическое выделение строк и строковые команды для поиска подстроки.

input.txt	Pum purum purum pum pum Pam param param pam pam it is a pumpkin
запуск	./filter input.txt pum
Вывод	Pum purum purum pum pum it is a pumpkin