Лабораторная работа №8.

**Тема: Функции в С.**

Задание №1.

**Постановка задачи:**

****

**Список идентификаторов:**

| **Переменная** | **Смысл** | **Тип** |
| --- | --- | --- |
| **sum** | Функция нахождения суммы | **float** |
| **summ** | Переменная для вычисления суммы в функции | **float** |
| **sA, sB, sC** | Длина массивов | **int** |
| **arr** | указатель на массив | **float** |
| **i** | счетчик цикла | **int** |
| **a, b, c** | Массивы вещественных чисел | **float** |
| **resA, resB ,resC, res** | Суммы | **float** |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

float sum(float \*arr, int s)

{

float summ=0;

for (int i=0; i<s; i++)

{

if (arr[i]>0) {summ+=arr[i];}

}

return summ;

}

int main()

{

int sA=6, sB=7, sC=8;

float a[6]={-1.2,-5.282,1.981,0,8.51,9.625},

b[7]={-8.6,-9.3,-5.12,2.8,0,0,1.115},

c[8]={0,0,1.8,188.2,5.108,-9.1,-1.2,-6.8};

float resA=0, resB=0, resC=0, res=0;

resA=sum(&a,sA);

printf("SUMMA A = %.3f",resA);

resB=sum(&b,sB);

printf("\nSUMMA B = %.3f",resB);

resC=sum(&c,sC);

printf("\nSUMMA C = %.3f",resC);

res=resA+resB+resC;

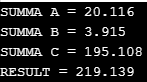
printf("\nRESULT = %.3f",res);

getchar();

return 0;

}

**Результат:**



Задание №2.

**Постановка задачи:**

****

**Список идентификаторов:**

| **Переменная** | **Смысл** | **Тип** |
| --- | --- | --- |
| **a, b** | Матрицы целых чисел | **int** |
| **max** | Максимальный элемент | **int** |
| **maxim** | Функция максимального элемента | **int** |
| **arr** | указатель на массив | **int** |
| **s, sAB** | размер массивов | **int** |
| **sum** | сумма элементов | **int** |
| **i** | счетчик цикла | **int** |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

int maxim(int \*arr, int s)

{

int max=arr[0];

for (int i=0; i<s; i++)

{

if (arr[i]>max) {max=arr[i];}

}

return max;

}

int main()

{

int sAB=8;

int a[8]={-1,-5,282,1,0,51,9,625},

b[8]={-8,6,-9,3,-512,28,0,115};

int sum=maxim(&a,sAB)+maxim(&b,sAB);

printf("\nRESULT = %.3d",sum);

getchar();

return 0;

}

**Результат:**



Задание №3.

**Постановка задачи:**

****

**Список идентификаторов:**

| **Переменная** | **Смысл** | **Тип** |
| --- | --- | --- |
| **arr** | указатель на массив | **float** |
| **s, sA, sV, sC** | размер массивов | **int** |
| **resA, resV, resC** | результаты | **float** |
| **k** | степень | **float** |
| **i** | счетчик цикла | **int** |
| **sg** | среднее геометрическое | **float** |
| **func** | функция нахождения среднего геометрического | **float** |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

float func(float \*arr, int s)

{

float sg=1, k=0;

for (int i=0; i<s; i++)

{

if (arr[i]>0) {sg\*=arr[i];k+=1;}

}

sg=pow(sg,1/k);

return sg;

}

int main()

{

int sA=6,

sV=7,

sC=8;

float a[6]={-1.2,-5.282,1.981,0,8.51,9.625},

v[7]={-8.6,-9.3,-5.12,2.8,0,0,1.115},

c[8]={0,0,1.8,188.2,5.108,-9.1,-1.2,-6.8};

float resA=0,

resV=0,

resC=0;

resA=func(&a,sA);

printf("SRGEOM A = %.3f",resA);

resV=func(&v,sV);

printf("\nSRGEOM V = %.3f",resV);

resC=func(&c,sC);

printf("\nSRGEOM C = %.3f",resC);

getchar();

return 0;

}

**Результат:**

  
Задание №4.

**Постановка задачи:  
**

**Список идентификаторов:**

| **Переменная** | **Смысл** | **Тип** |
| --- | --- | --- |
| **same** | Функция | **int** |
| **arr** | Массив с одинаковыми значениями | **int** |
| **S, K** | Сравниваемые массивы | **int** |
| **i, j, n, m, l** | Индексы массивов | **int** |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

void same(int S[3][3], int K[3][3], int arr[2])

{

int m=0, n=0, l=0;

while (n < 3)

{

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

if (S[n][l] == K[i][j])

{

arr[m] = S[n][l];

m++;

if (m > 2)

{

exit(0);

}

}

}

}

if (l < 2)

l++;

else {

n++;

l = 0;

}

}

}

int main() {

int S[3][3] = { 1, 2, 3,

4, 11, 6,

7, 8, 9 },

K[3][3] = { 10, 11, 12,

13, 14, 3,

15, 16, 17 },

arr[2];

same(S, K, arr);

printf("The same number is => %2d and %2d", arr[0], arr[1]);

getchar();

return 0;

}

**Результат:**



Задание №5.

**Постановка задачи:  
**

**Список идентификаторов:**

| **Переменная** | **Смысл** | **Тип** |
| --- | --- | --- |
| **arr** | указатель на массив | **int** |
| **v, g** | размеры массива | **int** |
| **sum** | функция суммы | **int** |
| **summ** | сумма | **int** |
| **min** | минимальный элемент | **int** |
| **i, j** | счетчики цикла | **int** |
| **vD, vG** | размеры массива | **int** |
| **d** | динамический массив | **int** |
| **res** | результат | **int** |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <malloc.h>

int sum(int \*\*arr, int v, int g)

{

int summ = 0, min = 0;

for (int j = 0; j < g; j++)

{

if (arr[0][j] > 0)

{

min += arr[0][j];

}

}

for (int i = 1; i < v; i++)

{

summ = 0;

for (int j = 0; j < g; j++)

{

if (arr[i][j] > 0) { summ += arr[i][j]; }

}

if (summ < min) { min = summ; }

}

return min;

}

int main()

{

int vD = 6, gD = 5;

srand(time(NULL));

int \*\*d;

d= (int\*\*)malloc(vD\*gD \* sizeof(int\*));

for (int i = 0; i < vD; i++)

{

printf("\n");

d[i] = (int\*)malloc(gD \* sizeof(int));

for (int j = 0; j < gD; j++)

{

d[i][j] = rand() % 20-9;

printf("%3d", d[i][j]);

}

}

int res = sum(d, vD, gD);

printf("\nRESULT = %d", res);

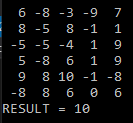
free;

getchar();

return 0;

}

**Результат:**



Задание №6.

**Постановка задачи:  
**

**Список идентификаторов:**

| **Переменная** | **Смысл** | **Тип** |
| --- | --- | --- |
| **d** | динамический массив | **int** |
| **res** | результат | **float** |
| **vD, vG** | размеры массива | **int** |
| **arr** | указатель на массив | **int** |
| **srg** | функция среднего геометрического | **float** |
| **geom** | среднее геометрическое | **float** |
| **pr** | произведение | **float** |
| **i, j** | счетчики цикла | **int** |
| **num** | номер столбца | **int** |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <math.h>

#include <malloc.h>

float srg(int \*\*arr, int num, int v, int g)

{

float geom = 1, pr = 1;

for (int i = 0; i < v; i++)

{

pr = pr\*arr[i][num];

}

geom = pow(pr, 1./ v);

return geom;

}

int main()

{

int vD = 3, gD = 5;

srand(time(NULL));

int \*\*d;

d = (int\*\*)malloc(vD\*gD \* sizeof(int\*));

for (int i = 0; i < vD; i++)

{

printf("\n");

d[i] = (int\*)malloc(gD \* sizeof(int));

for (int j = 0; j < gD; j++)

{

d[i][j] = rand() % 20 - 9;

printf("%3d", d[i][j]);

}

}

for (int j = 0; j < gD; j++)

{

float res = srg(d, j, vD, gD);

printf("\nRESULT = %.2f", res);

}

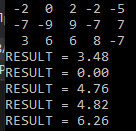
free(d);

getchar();

return 0;

}

**Результат:**



Задание №7.

**Постановка задачи:  
**

**Список идентификаторов:**

| **Переменная** | **Смысл** | **Тип** |
| --- | --- | --- |
| **arr** | указатель на массив | **int** |
| **v, g** | размеры массива | **int** |
| **i, j** | счетчики цикла | **int** |
| **vD, vG** | размеры массива | **int** |
| **d** | динамический массив | **int** |
| **miin** | функция минимального значения | **int** |
| **min** | минимальное значение | **int** |
| **res** | результат | **int** |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <math.h>

#include <malloc.h>

int miin(int \*\*arr, int v, int g)

{

int min;

for (int i = 0; i < v; i++)

{

min = arr[i][0];

for (int j = 0; j < g; j++)

{

if (arr[i][j] < min)

{

min = arr[i][j];

}

}

printf("\n min res in %d stroke = %d", i, min);

}

}

int main()

{

int vD = 3, gD = 5;

srand(time(NULL));

int \*\*d;

d = (int\*\*)malloc(vD\*gD \* sizeof(int\*));

for (int i = 0; i < vD; i++)

{

printf("\n");

d[i] = (int\*)malloc(gD \* sizeof(int));

for (int j = 0; j < gD; j++)

{

d[i][j] = rand() % 20 - 9;

printf("%3d", d[i][j]);

}

}

int res = miin(d, vD, gD);

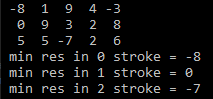
free;

getchar();

return 0;

}

**Результат:**

****

Задание №8.

**Постановка задачи:  
**

**Список идентификаторов:**

| **Переменная** | **Смысл** | **Тип** |
| --- | --- | --- |
| **n** | Число | **Int** |
| **count** | Счетчик | **Int** |
| **num** | Сравниваемое значение | **Int** |
| **prost** | Функция | **int** |
| **i** | Счетчик цикла | **int** |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

void prost(int n)

{

int count = 0, num = n;

for (int i = 1; i <= n;i++)

{

if (num%i == 0)

{

count++;

}

}

if (count > 2)

printf("Not simple");

else

{

printf("simple");

}

}

int main()

{

int n;

printf("You need to write a number ");

scanf("%d", &n);

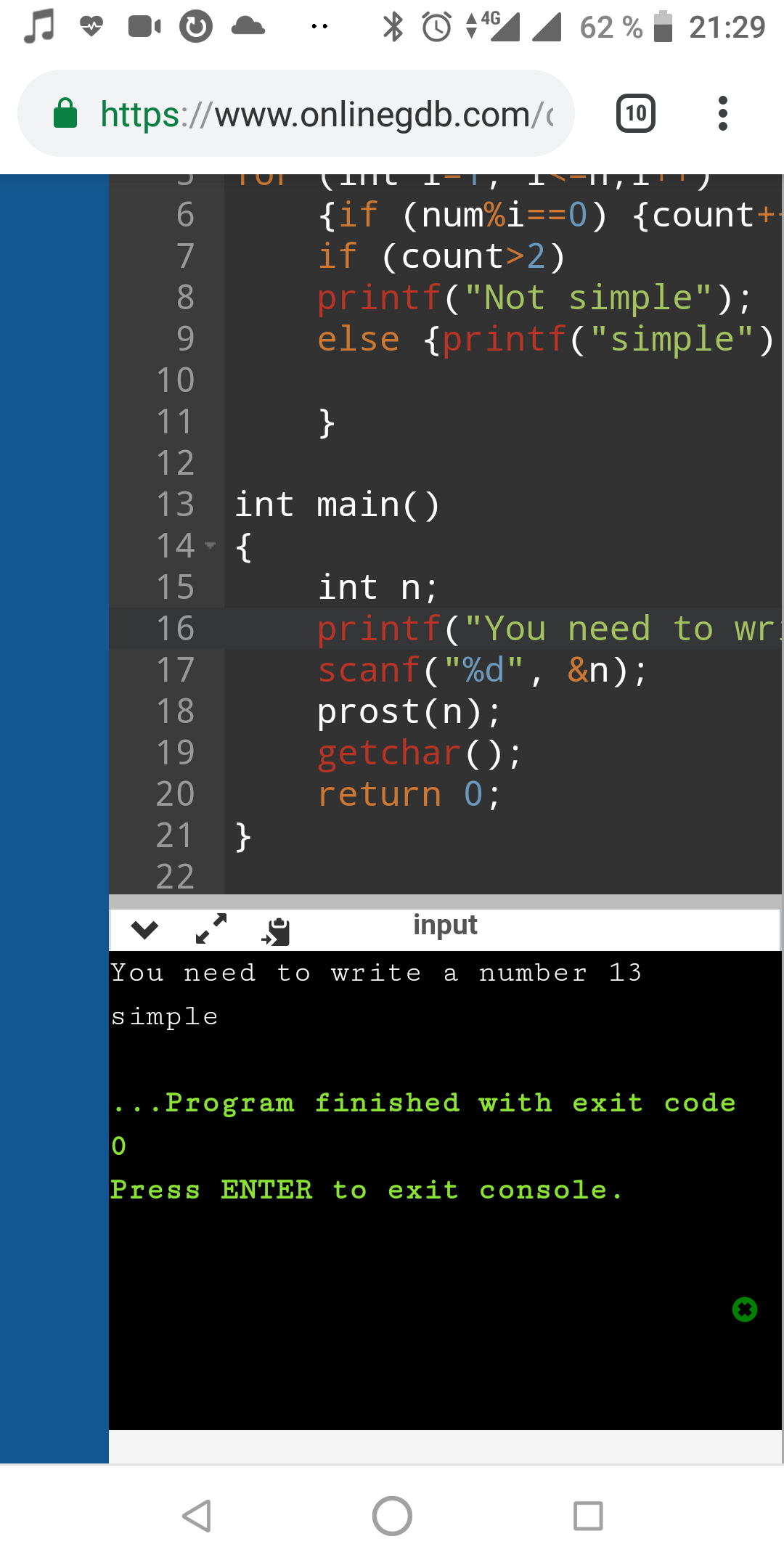
prost(n);

getchar();

return 0;

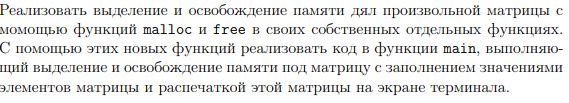
}

**Результат:**



Задание №9.

**Постановка задачи:**

****

**Список идентификаторов:**

| **Переменная** | **Смысл** | **Тип** |
| --- | --- | --- |
| **n, m** | размер матрицы | **int** |
| **m\_malloc** | функция создания матрицы | **int** |
| **free\_malloc** | функция освобождения памяти | **void** |
| **zap\_malloc** | функция заполнения матрицы | **void** |
| **arr** | матрица | **int** |
| **i, j** | счетчики цикла | **int** |
| **ch** | переменная выбора действия | **int** |
| **ob** | указатель | **void** |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <malloc.h>

int m\_malloc(int n, int m)

{

int \*\*arr;

arr = (int\*\*)malloc(n\*m \* sizeof(int\*));

for (int i = 0; i < n; i++)

{

arr[i] = (int\*)malloc(m \* sizeof(int));

}

return arr;

}

void zap\_malloc(int \*\*arr, int n, int m)

{

srand(time(NULL));

for (int i = 0; i < n; i++)

{

printf("\n");

for (int j = 0; j < m; j++)

{

arr[i][j] = rand() % 20 - 9;

printf("%3d", arr[i][j]);

}

}

printf("\n");

}

void free\_malloc(int \*\*arr, int n, int m)

{

for (int i = 0; i < n; i++)

{

free(arr[i]);

}

free(arr);

printf("\nMemory is clean\n");

}

int main()

{

int n, m;

printf("You need to write a size of matrix NxM = ");

scanf("%d%d", &n, &m);

int \*\*a= m\_malloc(n, m);

void(\*ob)(int \*arr, int n, int m);

ob = zap\_malloc;

int ch;

do {

printf("\nWhat do you wont to do with matrix:\n1.Fill with random numbers and display; \n2.Free up memory\n3.Leave this program?\nMy choice = ");

scanf("%d", &ch);

switch (ch)

{

case(1): ob = zap\_malloc;

break;

case(2): ob = free\_malloc;

break;

case(3): continue;

break;

}

ob(a, n, m);

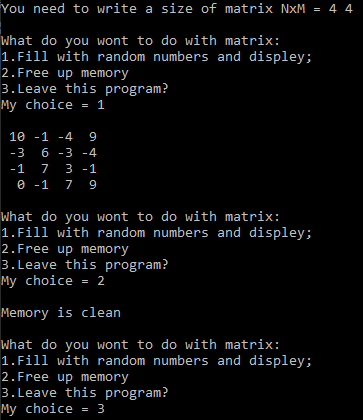
} while (ch!=3);

getchar();

return 0;

}

**Результат:**

  
Задание №10.

**Постановка задачи:**

Вычислить скорость электрона, вырванного из данного материала при фотоэффекте под воздействием излучения частоты ν. Необходимо исходить из простого энергетического соотношения где h ≈ 6.626×10−34 Дж·с – постоянная Планка, me ≈ 9.1×10−31 кг – масса электрона, A – работа выхода электрона из металлического материала, ν = c/λ – частота облучающего излучения, c = 2.998×108 м/с – скорость света в вакууме. Отсюда скорость электрона при вылете с поверхности металла определяется, как



Реализовать вычисление скорости электрона в виде отдельной функции. При вызове этой функции можно взять длину волны λ = 2.5 · 10−7 м. Работу выхода A при вызове функции указывать в электрон-вольтах (эВ). Внутри программы перевести эВ в Дж используя соответствие 1 эВ ≡ 1.602 × 10−19 Дж. Расчёты проводить в системе единиц СИ.

**Список идентификаторов:**

| **Переменная** | **Смысл** | **Тип** |
| --- | --- | --- |
| **h, m, n, y** | Константы | **define** |
| **A** | Работа выхода | **float** |
| **vi** | Частота облучающего излучения | **float** |
| **V** | Искомая скорость | **float** |
| **speed** | Функция нахождения скорости | **void** |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define h 6.626\*pow(10, -34)

#define m 9.1\*pow(10, -31)

#define c 2.998\*pow(10, 8)

#define y 2.5\*pow(10, -7)

void speed(float A)

{

float vi=0, V;

vi = c / y;

A = A\*1.602\*pow(10, -19);

V = sqrt((2./ m)\*(h\*vi - A));

printf("electron velocity = %f", V);

}

int main()

{

float A;

printf("You need to write A = ");

scanf("%f", &A);

speed(A);

getchar();

return 0;

}