**Тема 9. Функции**

**Т9.1.** Протабулировать функцию *f(x)=p∙sin(x)+q∙cos(x)* на интервале числовой оси от *a* до *b* с шагом *h.* Необходимо вычислить и вывести на экран значения функции в точках:



Коэффициенты p, q, координаты интервала и количество разбиений интервала *n*>0 вводятся с клавиатуры. Значения исходной функции в заданных точках вычисляются с помощью специально оформленной функции *ftrigs*. Функция имеет три формальных параметра для передачи в неё значений коэффициентов и координаты точки. В главную функцию возвращается значение типа float, вычисляемое в *ftrigs*.

#include <stdio.h>

#include <math.h>

float ftrigs(float p,float q,float x)

{float v;

v=p\*sin(x)+q\*cos(x);

return v;

}

int main()

{int n,i; float p,q,a,b,h,x,y;

printf("\n ‚Введите n,p,q,a и b\n");

scanf("%d%f%f%f%f",&n,&p,&q,&a,&b);

h=(b-a)/n;

for(i=0,x=a;i<=n;i++)

{y= ftrigs(p,q,x);x=x+h;

printf("y[%d]=%f\n",i,y);

}

return 1;

}

**Т9.3.** Представляемая программа показывает возможность передавать в функцию не только данные скалярного типа, но и целые массивы. Программа с помощью функции SclProd вычисляет скалярное произведение двух векторов.

#include <stdio.h>

#include <math.h>

#define Lim 100

float SclProd(int k,float x[],float y[])

{int i;float v;

for(i=0,v=0;i<k;i++)

v+=x[i]\*y[i];

return v;

}

int main()

{int n,i; float a[Lim],b[Lim],p;

printf("Размерность векторов n= \n");

scanf("%d",&n);

printf("Введите %d координат вектора a: \n",n);

for(i=0;i<=n;i++) scanf("%f",&a[i]);

printf("Введите %d координат вектора b: \n",n);

for(i=0;i<=n;i++) scanf("%f",&b[i]);

p=SclProd(n,a,b);

printf("Скалярное произведение векторов a и b равно %f",p);

return 1;

}

**Т9.4.** Рассмотренные ранее определения функций помещались перед вызывающей их функцией, т.е. перед главной. Стандарт языка Си предусматривает размещение определений функций и после вызывающей функции, однако требуется предварительное описание функций в виде прототипов.

**Задача.** В одномерном массиве вещественного типа найти максимальный элемент. Соответствующая программа содержит три функции InputMas, OutputMas и MaxMas. Их прототипы прописаны в программе до описания главной функции. Полное определение этих функций следует за главной функцией. Отметим, что первые две функции не возвращают результатов, и поэтому перед их именами используется тип void. Вызов таких функций совпадает с форматом вызова процедур. Другая особенность – в описании первой функции в качестве одного из формальных параметров объявлен указатель, что позволяет не только передать, но и возвратить из функции необходимое значение.

#include <stdio.h>

#define Lim 100

void InputMas(int \*, double []);

void OutputMas(int, double []);

double MaxMas(int, double []);

int main(){

double a[Lim], max;

int n=0;

InputMas(&n,a);

OutputMas(n,a);

max=MaxMas(n,a);

printf("\nmax = %.2lf\n",max);

return 1;

}

void InputMas(int\* n, double m[])

{int i,nn;

do{

printf("Введите 0<nn<=%d ",Lim);

scanf("%d",&nn);

}while(nn<=0 || nn>Lim);

printf("\nВводите элементы массива\n");

for(i=0;i<nn;i++){

printf("Введите элемент %d ",i+1);

scanf("%lf",&m[i]);

}

\*n=nn;

}

void OutputMas(int n, double m[])

{int i;

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

printf(" %.2lf ",m[i]);

}

double MaxMas(int n, double m[]){

double max; int i;

max=m[0];

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

if(m[i]>max) max=m[i];

return max;

}

**Т9.5.** Аналогично, в качестве формальных параметров функций могут выступать и двумерные массивы.

**Задача.** В двумерном массиве вещественного типа найти максимальный элемент. Соответствующая программа содержит три функции InputMas, OutputMas и MaxMas. Их прототипы прописаны в программе до описания главной функции. Функции имеют по три формальных параметра. Два параметра для первой функции являются указателями.

#include <stdio.h>

#define Lim 10

void InputMas(int\*, int\*, double [][Lim]);

void OutputMas(int, int, double [][Lim]);

double MaxMas(int, int, double [][Lim]);

int main()

{ double a[Lim][Lim], max;

int i,n=0,m=0;

InputMas(&n,&m,a);

OutputMas(n,m,a);

max=MaxMas(n,m,a);

printf("\nmax = %.2lf\n",max);

return 1;

}

void InputMas(int\* n, int\* m, double a[][Lim])

{ int i,j,nn,mm;

double temp;

do{

printf("Введите 0<n<=%d ",Lim);

scanf("%d",&nn);

}while(nn<=0 || nn>Lim);

do{

printf("Введите 0<m<=%d ",Lim);

scanf("%d",&mm);

}while(mm<=0 || mm>Lim);

printf("\nВведите элементы массива\n");

for(i=0;i<nn;i++)

for(j=0;j<mm;j++){

printf("Введите элемент с индексами %d %d ",i,j);

scanf("%lf",&temp);a[i][j]=temp;

}

\*n=nn;

\*m=mm;

}

void OutputMas(int n, int m, double a[][Lim])

{int i,j;

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

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

printf("%5.1f",a[i][j]);

printf("\n");

}

}

double MaxMas(int n, int m, double a[][Lim])

{int i,j;

double max;

max=a[0][0];

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

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

if(a[i][j]>max) max=a[i][j];

return max;

}

**Т9.6.** Рассмотрим задачу о вычислении биномиального коэффициента:

,

где *n ≥m≥0*; *n, m* – целые числа.

В данном примере программы для вычисления факториала оформлена своя функция, и биномиальный коэффициент выражается посредством этой функции при разных значениях фактического параметра. В отличие от предыдущих примеров здесь реализуется рекурсия, т.е. в теле оформляемой функции идёт обращение к этой же функции, но при другом значении параметра.

#include <iostream>

long fact(int k)

{if (k<0) return 0;

if (k==0) return 1;

return k\*fact(k-1);}

int main ()

{int n,m,nmc,nm;

cout <<"n=";cin>>n;

cout<<"m=";cin>>m;

nm=n-m;

nmc=fact(n)/fact(m)/fact(nm);

cout<<"nmc="<<nmc;

return 1;

}

**Тема 10. Структуры данных**

**Т10.1.** Наряду с указателями, массивами и функциями структура является также производным типом, объединяющим в единое целое множество, в общем случае, разнотипных элементов. Такой тип данных имеет многочисленные применения.

**Задача.** Составить программу с вводом и выводом данных для структуры, содержащей поля: название направления; количество групп; общее число студентов; количество преуспевающих студентов.

#include <stdio.h>

#define LimS 100

#define LimG 10

struct Tutor //Объявляется структурный тип

//данных под именем Tutor

{char name\_f[]; //название направления

int n\_group; //число групп

int n\_stud; //общее число студентов

int n\_estud; //число преуспевающих студентов

};

int main(){

struct Tutor T;//Объявление переменной типа

//структура

printf("Введите название направления\n");

scanf("%s",T.name\_f);

do{

printf("Введите количество групп 0<g<=%d ",LimG);

scanf("%d",&T.n\_group);

}while(T.n\_group<=0 || T.n\_group>LimG);

do{

printf("Введите общее количество студентов 0<n<=%d ",LimS);

scanf("%d",&T.n\_stud);

}while(T.n\_stud<=0 || T.n\_stud>LimS);

do{

printf("Введите количество преуспевающих студентов 0<=ne<=%d ",LimS);

scanf("%d",&T.n\_estud);

}while(T.n\_estud<0 || T.n\_estud>LimS);

printf("Направление: %s\n",T.name\_f);

printf("Число групп: %d\n",T.n\_group);

printf("Число студентов: %d\n",T.n\_stud);

printf("Число преуспевающих студентов: %d\n",T.n\_estud);

return 1;

}

**Т10.2.** В данном примере рассматривается целый массив из элементов типа структуры Tutor из предыдущего примера. Для обработки данных массива структур определены три функции InputS, OutputS и ProzentS, каждая с двумя формальными параметрами. В качестве первого параметра фигурируют массивы структур. Первые две функции предназначены, соответственно, для ввода и вывода данных массива. С помощью третьей функции вычисляется процент преуспевающих студентов относительно общего количества студентов.

#include <stdio.h>

#define LimS 100

#define LimG 10

#define LimT 5

struct Tutor{

char name\_f[]; //название направления

int n\_group; //число групп

int n\_stud; //общее число студентов

int n\_estud;//число преуспевающих студентов

};

void InputS(struct Tutor [], int\*);

void OutputS(struct Tutor [], int);

void ProzentS(struct Tutor [], int);

int main(){

struct Tutor ta[LimT];//Объявление массива

//структур

int t;

InputS(ta,&t);

OutputS(ta,t);

ProzentS(ta,t);

return 1;

}

void InputS(struct Tutor t[LimT], int \* n)

{int i,nn;

do{

printf("Введите размерность массива 0<n<=%d ",LimT);

scanf("%d",&nn);

}while(nn<=0 || nn>LimT);

printf("Ввод данных по запросу программы:\n");

for(i=0;i<nn;i++)

{ printf("\nВведите данные для направления %d\n",i+1);

printf("Введите название (например NP | NI | NK | NB | NN) \n");

scanf("%s",t[i].name\_f);

do{

printf("Введите количество групп 0<g<=%d ",LimG);

scanf("%d",&t[i].n\_group);

}while(t[i].n\_group<=0 || t[i].n\_group>LimG);

do{

printf("Введите общее количество студентов 0<n<=%d ",LimS);

scanf("%d",&t[i].n\_stud);

}while(t[i].n\_stud<=0 || t[i].n\_stud>LimS);

do{

printf("Введите количество преуспевающих студентов 0<=es<=%d ",LimS);

scanf("%d",&t[i].n\_estud);

}while(t[i].n\_estud<0 || t[i].n\_estud>LimS);

}

\*n=nn;

}

void OutputS(struct Tutor t[LimT], int n){

int i;

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

printf("\Название направления %d: %s\n",i+1,t[i].name\_f);

printf("Кол-во групп: %d\n",t[i].n\_group);

printf("Общее кол-во студентов: %d\n",t[i].n\_stud);

printf("Кол-во преуспевающих студентов: %d\n",t[i].n\_estud);

}

}

void ProzentS(struct Tutor t[LimT], int n)

{int i,Sn,Se; float p;

Sn=0;Se=0;

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

{Sn+=t[i].n\_stud; Se+=t[i].n\_estud;}

p=100.0\*Se/Sn;

printf("%2.2f%%\n",p);

}