Одномерные массивы

1. Определение: Массив – упорядоченная совокупность данных одного типа

2. Описание массива

<тип> <имя массива>(<размер массива>);

Примеры:

int a[10]; float b[40];

Нумерация элементов:

a[0],a[1],…,a[9]

Расположение в памяти – в последовательных ячейках без пропусков между элементами

a[0]	a[1]	a[2]	a[3]	…	a[8]	a[9]

Обращение к отдельным элементам массива:

a[i] – элемент массива с индексом I

3. Ввод одномерных массивов – обычно в цикле for

Операции с массивами

- for(int i=0;i<10;i++)
- cin>>a[i];
- В строке ввода элементы могут располагаться в любом порядке: - построчно или в столбик
- 4. Вывод одномерных массивов – обычно в цикле for()
- for(int i=0;i<10;i++)
- cout<<a[i]<<“ “;
- Вид выводимого массива задается программистом. В описанном выше примере элементы выводятся в строку. Между элементами ставятся пробелы.
- Ниже рассмотрен вывод массива в столбик
- for(int i=0;i<10;i++)
- cout<<a[i]<<endl;
- 5.Примеры решения задач
- **Задача 1** В одномерном целочисленном массиве a[10] определить сумму элементов, делящихся на число K

Примеры задач

- без остатка, и поставит эту сумму на место максимального элемента массива
- #include “stdafx.h”
- #include “iostream”
- using namespace std;
- int main()
- {
- int a[10], K;
- cout<<“Input K: “;
- cin>>K;
- cout<<“Input array a[10]”<<endl;
- for(int i=0;i<10;i++)
- cin>>a[i];
- int s=0, max=a[0],imax=0;

Примеры задач

```
for(int i=0;i<10;i++)
{
  if(a[i]>max)
  {
    max=a[i];
    imax=i;
   }
 if(a[i]%K==0)
  s=s+a[i];
}
 a[imax]=s;
 cout<<“new array”<<endl;
 for(int i=0;i<10;i++)
  cout<<a[i]<<“  “;
 system(“pause”);
```

Примеры задач

- return 0;
- }
- **Задача 2.** Четные элементы одномерного целочисленного массива a[10] переписать в массив c, а нечетные – в массив b.
- Основная особенность – использование счетчиков. Счетчик – целая переменная, начальное значение которой равно 0, а в процессе выполнения программы ее значение всякий раз увеличивается на 1 при наступлении заданной ситуации
- #include “stdafx.h”
- #include “iostream”
- using namespace std;
- int main()
- {
- int a[10],b[10],c[10];
- int kb=0,kc=0;

Примеры задач

```
cout<<“Input array a[10]”<<endl;
for(int i=0;i<10;i++)
 cin>>a[i];
for(int i=0;i<10;i++)
{
if(a[i]%2==0)
 {
  c[kc]=a[i];
  kc++;
 }
if(a[i]%2!=0)
 {
  b[kb]=a[i];
  kb++;
 }
}
```

Примеры задач

- cout<<“array b”<<endl;
- for(int i=0;i<kb;i++)
- cout<<b[i]<<“ “;
- cout<<endl;
- cout<<“array c”<<endl;
- for(int i=0;i<kc;i++)
- cout<<c[i]<<“ “;
- system(“pause”);
- return 0;
- }
- **Задача 3** В массиве a[10] поменять местами 1-ый отрицательный элемент и минимальный элемент этого массива. Если отрицательных элементов нет, выдать соответствующее сообщение.
- Отличительной особенностью задача является наличие флагов – переменных, принимающих значение 1, если заданная ситуация произошла, и 0, если не произошла.

Примеры задач

```
• #include “stdafx.h”
• #include “iostream”
•  using namespace std;
•  int main()
• {
•  int a[10],b[10];
•  cout<<“Input array a[10]”<<endl;
•  for(int i=0;i<10;i++)
•   cin>>a[i];
•  int min=a[0],imin=0;
•  int fl=0, iotr;
•  for(int i=0;i<10;i++)
•  {
•   if(a[i]<min)
```

Примеры задач

```
{
min=a[i];
imin=i;
}
if(a[i]<0 && fl==0)
{
iotr = i;
fl=1;
}
}
if(fl>0)
{
a[imin]=a[iotr];
a[iotr]=min;
cout<<“new array ”<<endl;
for(int i=0;i<10;i++)
cout<<a[i]<<“  “;
}
```

Примеры задач

- else
- cout<<“No negative elements”;
- system(“pause”);
- return 0;
- }
- **Задача 4.** В целочисленном массиве a[10] определить наиболее часто встречающийся элемент. Получить новый массив m[10] путем поэлементного умножения массива a[10] на этот элемент.
- #include “stdafx.h”
- #include “iostream”
- using namespace std;
- int main()
- {
- int a[10],m[10];
- cout<<“Input array a[10]”<<endl;
- for(int i=0;i<10;i++)
- cin>>a[i];

Примеры задач

```
int max=0,amax;
for(int i=0;i<10;i++)
{
 int k=0;
 for(int j=0;j<10;j++)
 if(a[i]==a[j])
  k++;
 if(k>max)
 {
  max=k;
  amax = a[i];
 }
}
for(int i=0;i<10;i++)
 m[i]=a[i]*amax;
```

Примеры задач

- cout<<“new array ”<<endl;
- for(int i=0;i<10;i++)
- cout<<m[i]<<“ “;
- system(“pause”);
- return 0;
- }
- **Замечание.** Общая рекомендация: операции ввода и вывода реализуются отдельно от операций, входящих в состав алгоритма решения задачи.