



МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение
высшего образования*

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Отчет по выполнению практического занятия № 1

Тема. Технология реализации алгоритмов с
использованием функций (процедурное программирование) в
заданиях дисциплины. структурами статические и
динамические массивы

Дисциплина Структуры и алгоритмы обработки данных

Выполнил студент
группа

Ло В.Х
ИНБО-15-20

Москва 2021

1 Задание 1

Разработать программу выполнения операций над статическим массивом целых чисел. Размер массива 1000 элементов. Предусмотреть ввод значения n – текущий размер массива. Разработать операции для управления массивом и реализовать их функциями. Функции должны принимать входные данные через параметры и возвращать результат, если этого требует алгоритм операции. Требования к содержанию кода функций:

Для каждой функции указать предусловие и постусловие.

Функция принимает на вход корректные данные. Контроль на некорректность выполнить до вызова функции, о чем указать в предусловии. Если пользователь все-же передал некорректные данные, вернуть код завершения и об этом указать в постусловии функции.

В функциях, которые выполняют вычисления не должно быть действий по вводу и выводу.

Примечание. Предусловие определяет какие исходные данные поступают в алгоритм. Постусловие определяет все возможные результаты работы алгоритма.

- 1) Разработать функцию ввода массива из n элементов с клавиатуры.
- 2) Разработать функцию вывода массива из n на монитор.
- 3) Разработать функцию заполнения массива из n элементов, используя датчик случайных чисел.
- 4) Разработать программу и протестировать работу функций.
- 5) Разработать функцию поиска первого вхождения значения в массив. Разработать набор тестов для данной операции.
- 6) Разработать функцию нахождения индекса первого отрицательного числа в массиве.
- 7) Разработать функцию поиска всех вхождений в массив.
- 8) Разработать функцию вставки нового значения в заданную позицию массива. Примечание. Рассмотреть на примере массива алгоритм вставки нового значения..

9) Разработать функцию алгоритма удаления со сжатием из массива значения в заданной позиции, сохраняя порядок следования остальных элементов.

10) Разработать функцию алгоритма удаления со сжатием всех вхождений заданного значения из массива. Сложность алгоритма $O(n^2)$.

Примечание. Рассмотреть на примере массива алгоритм удаления со сжатием значения. Показать появление квадратичной зависимости времени от n .

11) Разработать функцию алгоритма удаления со сжатием всех вхождений заданного значения из массива. Сложность алгоритма $O(n)$.

Примечание. Рассмотреть на примере массива алгоритм удаления со сжатием значения. Показать появление линейной зависимости времени от n .

2 Задание 2

Реализовать операции вставки и удаления значений в динамическом массиве, модифицируя соответствующие функции статического массива. Использовать функцию `realloc` модуля `malloc` при изменении размера массива. Выполнить тестирование операций изменения размера массива. Применить к массиву


```

void Insert(int* x, int &n, int h , int g){
    n++;
    for(int i=n-1;i>g;i--)
        x[i]=x[i-1];

    x[g]=h;
}
void delete_x(int* x, int &n, int idx)
{
    if (idx >= 0 && idx < n)
    {
        for (int i = idx; i < n - 1; i++)
            x[i] = x[i + 1];
        n--;
    }
}
void delete_the_same_element_Onn(int* x, int &n,int d)
{
    for(int i = 0; i < n; i++)
    {
        if(x[i]==d)
            for(int j = i; j < n-1; j++)
            {
                x[j]=x[j+1];
            }
        n--;
    }
}
void delete_the_same_element_On(int* x, int &n,int d)
{
    int j=0;
    for(int i=0;i<n;i++)
    {
        x[j]=x[i];
        if(x[i]!=d)
        {
            j++;
        }
    }
    n=j;
}

```

2) Код основной программы

```
const int N = 1000;
int main(){
    int a[N];
    int n,k,h,g,idx,d,ch;
    do
    {
        cout << "Enter n: ";
        cin >> n;
        if(n<=0 || n>N)
            cout<<"error 0<n<=N."<<endl;
    }while(n<=0 || n>N);
    cout << "\n\t\tfunctions"<<endl;
    while (1)
    {
        cout<<"\n1. input array with keyboard " << endl;
        cout<<"2. input array with random " << endl;
        cout<<"3. the first position of the element " << endl;
        cout<<"4. Develop a function to find all occurrences of an array." << endl;
        cout<<"5. find negative numbers " << endl;
        cout<<"6. the number to add " << endl;
        cout<<"7. positions to be deleted " << endl;
        cout<<"8. positions to be deleted with 1 for " << endl;
        cout<<"9. positions to be deleted with 2 for " << endl;
        cout<< "Please enter: ";
        cin >> ch;
        if (cin.fail())
        {
            cin.clear();
            cin.ignore(32767, '\n');
        }

        if(ch==1)
        {
            input(a, n);
            output(a, n);
        }
        if(ch==2)
        {
            Random(a, n);
            output(a, n);
        }
        if(ch==3)
        {
            output(a, n);
            cout<<"\nenter key:";
            cin>>k;
            cout<<"the first position of the element: "<<Finde_index(a,n,k)<<endl;
        }
        if(ch==4)
        {
            Count(a,n);
        }
        if(ch==5)
        {
            int p = find_negative_numbers(a,n);
            if(p!=-1)
            {
                cout<<"find negative numbers"<<p<<endl;
            }
            else
            {
                cout<<"The first negative number was not found."<<endl;
            }
        }
        if(ch==6)
```

```

}
if(ch==6)
{
cout<<"\nenter the number to add: ";
cin>>h;
cout<<"enter the location: ";
cin>>g;
Insert(a,n,h,g);
cout<<"array after add: "<<endl;
output(a, n);
}
if(ch==7)
{
cout << "\npositions to be deleted: ";
cin >> idx;
delete_x(a, n, idx);
cout << "marray after deletion: " << endl;
output(a, n);
}
if(ch==8)
{
cout << "\npositions to be deleted with 1 for: ";
cin >>d;
delete_the_same_element_On(a,n,d);
cout << "\narray after deleting the same element:\n";
output(a, n);
}
if(ch==9)
{
cout << "\npositions to be deleted with 2 for: ";
cin >>d;
delete_the_same_element_Onn(a,n,d);
cout << "\narray after deleting the same element:\n";
output(a, n);
}
else
{
cout<<"wrong input!\nplease input again."<<endl;
}
}
system("pause");
return 0;
}

```

3) Тест для каждой операции

Название операции

Номер теста	Входные данные	Ожидаемый результат
1- Заполнение массива: с клавиатуры и вывод массива	5 1 3 5 6 5 7	a[0] = 3 a[1] = 5 a[2] = 6 a[3] = 5 a[4] = 7
2- Заполнение массива: датчиком случайных чисел и вывод массива	5 2	(Random number with srand(time(NULL)))
3 - Найти индекс первого вхождения заданного значения в массиве	3 6	Array have value 1 at position: 2
4 - Найти количество вхождений заданного значения в массиве (поиск всех вхождений)	4 5	Number of occurrences of 2 in array is : 2
5 - индекса первого отрицательного числа в массиве.	5	The first negative number was not found.
6 - Вставить значение в заданную позицию (контроль позиции в основной программе).	6 3 9	a[0] = 3 a[1] = 5 a[2] = 6 a[3] = 9 a[4] = 5 a[5] = 7
7 - Удалить значение из заданной позиции	7 3	a[0] = 3 a[1] = 5 a[2] = 6 a[3] = 5 a[4] = 7
8 - Удалить все значения равные заданному значению, сохраняя порядок следования элементов: алгоритмом с одним циклом	8 5	a[0] = 3 a[1] = 6 a[2] = 7
9 - Удалить все значения равные заданному значению, сохраняя порядок следования элементов: алгоритмом с двумя циклами	9 6	a[0] = 3 a[1] = 7

Скриншот выполнения теста.

1. Заполнение массива: с клавиатуры и вывод массива:

```
Enter n: 5

functions

1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 1
input element of array with keyboard.
Enter array x[0] : 3
Enter array x[1] : 5
Enter array x[2] : 6
Enter array x[3] : 5
Enter array x[4] : 7
3 5 6 5 7
```

2. Заполнение массива: датчиком случайных чисел и вывод массива:

```
1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 2
input element of array with random.
72 29 61 66 35
```

3. Найти индекс первого вхождения заданного значения в массиве.

```
1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 3
3 5 6 5 7
enter key:6
the first position of the element: 2
```


4. Найти количество вхождений заданного значения в массиве

```
1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 4

enter the number to be counted: 5
number 5 appear 2 times
```

5. Индекса первого отрицательного числа в массиве.

```
1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 5
The first negative number was not found.
```

6. Вставить значение в заданную позицию.

```
1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 6

enter the number to add: 9
enter the location: 3
array after add:
3 5 6 9 5 7
```

7. Удалить значение из заданной позиции.

```
1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 7

positions to be deleted: 3
array after deletion:
3 5 6 5 7
```

8. удаление со сжатием всех вхождений заданного значения из массива.
Сложность алгоритма $O(n)$.

```
1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 8

positions to be deleted with 1 for: 5

array after deleting the same element:
3 6 7
```

8. удаление со сжатием всех вхождений заданного значения из массива.
Сложность алгоритма $O(n^2)$.

```
1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 9
3 6 7
positions to be deleted with 2 for: 6

array after deleting the same element:
3 7
```

Задание 2

1) Код определений функций (файл с расширением cpp)

```
#include<iostream>
#include <stdlib.h>
#include <time.h>
using namespace std;
void input(int* x, int n)
{
    cout<<"input element of array with keyboard."<<endl;
    for (int i = 0; i < n; i++)
    {
        cout << "Enter array x[" << i << "] : ";
        cin >> x[i];
    }
}
void Random(int* x, int n)
{
    srand(time(NULL));
    for(int i=0;i<n;i++)
    {
        x[i]=rand()%100;
    }
}
void output(int* x, int n)
{
    for (int i = 0; i < n; i++)
        cout << x[i] << " ";
}
int Finde_index(int* x, int n, int k)
{
    int m;
    for(int i=0;i<n;i++)
    {
        if(x[i]==k)
            return i;
    }
    return -1;
}
int find_negative_numbers(int* x,int n)
{
    for(int i=0;i<n;i++)
        if(x[i]<0)
            return i;
    return -1;
}
int search(int* x, int n, int c)
{
    for(int i=0;i<n;i++)
    {
        if(x[i]==c)
            return i;
    }
    return -1;
}
void Count(int* x, int n){
    int count =0, c;

    cout<<"\nenter the number to be counted: ";
    cin>>c;
    for(int i=0;i<n;i++){
        if(x[i]==c)
            count ++;
    }

    cout<<"number " <<c<<" appear " <<count<<" times"<<endl;
}
```

```

void Insert(int* &x, int &n, int h , int g)
{
    x= (int*)realloc(x,(n+1)*sizeof(int));
    n++;
    for(int i=n-1;i>g;i--)
    {
        x[i]=x[i-1];
    }
    x[g]=h;
}

void delete_x(int* &x, int &n, int idx)
{
    if (idx >= 0 && idx < n)
    {
        for (int i = idx; i < n - 1; i++)
            x[i] = x[i + 1];
        n--;
    }
    x= (int*)realloc(x,n*sizeof(int));
}

void delete_the_same_element_Onn(int* &x, int &n,int d)
{
    for(int i = 0; i < n; i++)
    {
        if(x[i]==d)
        {
            for(int j = i; j < n-1; j++)
            {
                x[j]=x[j+1];
            }
            n--;
        }
    }
    x= (int*)realloc(x,n*sizeof(int));
}

void delete_the_same_element_On(int* x, int &n,int d)
{
    int j=0;
    for(int i=0;i<n;i++)
    {
        x[j]=x[i];
        if(x[i]!=d)
        {
            j++;
        }
    }
    n=j;
    x= (int*)realloc(x,n*sizeof(int));
}

```

2) Код основной программы

```
const int N = 1000;
int main(){
    int a[N];
    int n,k,h,g,idx,d,ch;
    int* x = new int[n];
    do
    {
        cout << "Enter n: ";
        cin >> n;
        if(n<=0 || n>N)
            cout<<"error 0<n<=N."<<endl;
    }while(n<=0 || n>N);
    cout << "\n\t\tfunctions"<<endl;
    while (1)
    {
        cout<<"\n1. input array with keyboard " << endl;
        cout<<"2. input array with random " << endl;
        cout<<"3. the first position of the element " << endl;
        cout<<"4. Develop a function to find all occurrences of an array." << endl;
        cout<<"5. find negative numbers " << endl;
        cout<<"6. the number to add " << endl;
        cout<<"7. positions to be deleted " << endl;
        cout<<"8. positions to be deleted with 1 for " << endl;
        cout<<"9. positions to be deleted with 2 for " << endl;
        cout<< "Please enter: ";
        cin >> ch;
        if (cin.fail())
        {
            cin.clear();
            cin.ignore(32767, '\n');
        }

        if(ch==1)
        {
            input(x, n);
            output(x, n);
        }
        else if(ch==2)
        {
            Random(x, n);
            output(x, n);
        }
        else if(ch==3)
        {
            output(x, n);
            cout<<"\nenter key:";
            cin>>k;
            cout<<"the first position of the element: "<<Finde_index(x,n,k)<<endl;
        }
        else if(ch==4)
        {
            Count(x,n);
        }
        else if(ch==5)
        {
            int p = find_negative_numbers(x,n);
            if(p!=-1)
            {
                cout<<"find negative numbers"<<p<<endl;
            }
            else
            {
                cout<<"The first negative number was not found."<<endl;
            }
        }
    }
}
```

```

}
else if(ch==6)
{
    output(x, n);
    cout<<"\nenter the number to add: ";
    cin>>h;
    cout<<"enter the location: ";
    cin>>g;
    Insert(x,n,h,g);
    cout<<"array after add: "<<endl;
    output(x, n);
}
else if(ch==7)
{
    output(x, n);
    cout << "\npositions to be deleted: ";
    cin >> idx;
    delete_x(x, n, idx);
    cout << "marray after deletion: " << endl;
    output(x, n);
}
else if(ch==8)
{
    cout << "\npositions to be deleted with 1 for: ";
    cin >>d;
    delete_the_same_element_On(x,n,d);
    cout << "\narray after deleting the same element:\n";
    output(x, n);
}
else if(ch==9)
{
    output(x, n);
    cout << "\npositions to be deleted with 2 for: ";
    cin >>d;
    delete_the_same_element_Onn(x,n,d);
    cout << "\narray after deleting the same element:\n";
    output(x, n);
}
else
{
    cout<<"wrong input!\nplease input again."<<endl;
}
}
system("pause");
return 0;
}

```

3) Тест для каждой операции

Название операции

Номер теста	Входные данные	Ожидаемый результат
1- Заполнение массива: с клавиатуры и вывод массива	5 1 1 3 5 5 7	a[0] = 1 a[1] = 3 a[2] = 5 a[3] = 5 a[4] = 7
2- Заполнение массива: датчиком случайных чисел и вывод массива	6 2	(Random number with srand(time(NULL)))
3 - Найти индекс первого вхождения заданного значения в массиве	3 7	Array have value 1 at position: 4
4 - Найти количество вхождений заданного значения в массиве (поиск всех вхождений)	4 5	Number of occurrences of 2 in array is : 2
5 - индекса первого отрицательного числа в массиве.	5	The first negative number was not found.
6 - Вставить значение в заданную позицию (контроль позиции в основной программе).	6 6 3	a[0] = 1 a[1] = 3 a[2] = 5 a[3] = 6 a[4] = 5 a[5] = 7
7 - Удалить значение из заданной позиции	7 3	a[0] = 1 a[1] = 3 a[2] = 5 a[3] = 5 a[4] = 7
8 - Удалить все значения равные заданному значению, сохраняя порядок следования элементов: алгоритмом с одним циклом	8 5	a[0] = 1 a[1] = 3 a[2] = 7
9 - Удалить все значения равные заданному значению, сохраняя порядок следования элементов: алгоритмом с двумя циклами	9 3	a[0] = 1 a[1] = 7

Скриншот выполнения теста.

1. Заполнение массива: с клавиатуры и вывод массива:

```
Enter n: 5

          functions

1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 1
input element of array with keyboard.
Enter array x[0] : 1
Enter array x[1] : 3
Enter array x[2] : 5
Enter array x[3] : 5
Enter array x[4] : 7
1 3 5 5 7
```

2. Заполнение массива: датчиком случайных чисел и вывод массива:

```
Enter n: 5

          functions

1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 2
3 33 84 2 12
```

3. Найти индекс первого вхождения заданного значения в массиве.

4.

```
Enter array x[0] : 1
Enter array x[1] : 3
Enter array x[2] : 5
Enter array x[3] : 5
Enter array x[4] : 7
1 3 5 5 7
1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 3
1 3 5 5 7
enter key:7
the first position of the element: 4
```


5. Найти количество вхождений заданного значения в массиве

```
1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 4

enter the number to be counted: 5
number 5 appear 2 times
```

6. Индекса первого отрицательного числа в массиве.

7.

```
1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 5
The first negative number was not found.
```

6. Вставить значение в заданную позицию.

```
1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 6
1 3 5 5 7
enter the number to add: 6
enter the location: 3
array after add:
1 3 5 6 5 7
```

7. Удалить значение из заданной позиции.

```
1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 7
1 3 5 6 5 7
positions to be deleted: 3
array after deletion:
1 3 5 5 7
```

8. удаление со сжатием всех вхождений заданного значения из массива.
Сложность алгоритма $O(n)$.

```
1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 8

positions to be deleted with 1 for: 5

array after deleting the same element:
1 3 7
```

9. удаление со сжатием всех вхождений заданного значения из массива.
Сложность алгоритма $O(n^2)$.

```
1. input array with keyboard
2. input array with random
3. the first position of the element
4. Develop a function to find all occurrences of an array.
5. find negative numbers
6. the number to add
7. positions to be deleted
8. positions to be deleted with 1 for
9. positions to be deleted with 2 for
Please enter: 9
1 3 7
positions to be deleted with 2 for: 3

array after deleting the same element:
1 7
```

1.2 Ответить на вопросы в отчете

Выполнить опрос на знание понятий:

- 1) Что такое указатель?
- 2) Какую структуру называем динамической?
- 3) Где хранится динамическая структура данных?
- 4) Где хранится статический массив?
- 5) Когда создается статический массив (структура)?
- 6) Можно ли увеличить/уменьшить физический объем статического массива?
- 7) Можно ли увеличить/уменьшить физический объем динамического массива?

- 1) Что такое указатель?

Ответ:

Указатель – переменная, значением которой является адрес ячейки памяти. То есть указатель ссылается на блок данных из области памяти, причём на самое его начало. Указатель может ссылаться на переменную или функцию. Для этого нужно знать адрес переменной или функции. Так вот, чтобы узнать адрес конкретной переменной в C++ существует унарная операция взятия адреса &. Такая операция извлекает адрес объявленных переменных, для того, чтобы его присвоить указателю.

- 2) Какую структуру называем динамической?

Ответ:

Динамические структуры данных — это любая структура данных, занимаемый объем памяти которой не является фиксированным. Иными словами, в подобной структуре может храниться как два, пять, двадцать элементов, так и одно большое ничего. Размер подобной структуры ограничен только объемом оперативной памяти компьютера.

- 3) Где хранится динамическая структура данных?

Ответ:

Динамическая структура данных хранится в куче

- 4) Где хранится статический массив?

Ответ:

Статический массив хранится в стеке

- 5) Когда создается статический массив (структура)?

Ответ:

Статическая память хранит глобальные переменные и константы; размер определяется при компиляции; размер памяти остается неизменным в процессе работы программы.

- 6) Можно ли увеличить/уменьшить физический объем статического массива?

Ответ: нет, этого нельзя сделать. Отсюда и название "статичный".

7) Можно ли увеличить/уменьшить физический объем динамического массива?
Ответ:

Динамическим называется массив, размер которого может изменяться во время исполнения программы. Возможность изменения размера отличает динамический массив от статического, размер которого задаётся на момент компиляции программы