جامعة الملك سعود
King Saud University

Arrays – Part 2

Lecture 12

Outline

- Using Arrays
- Arrays in Functions
- Searching Partially Filled Arrays
- Sorting Arrays
- Arrays and Classes

Using Arrays

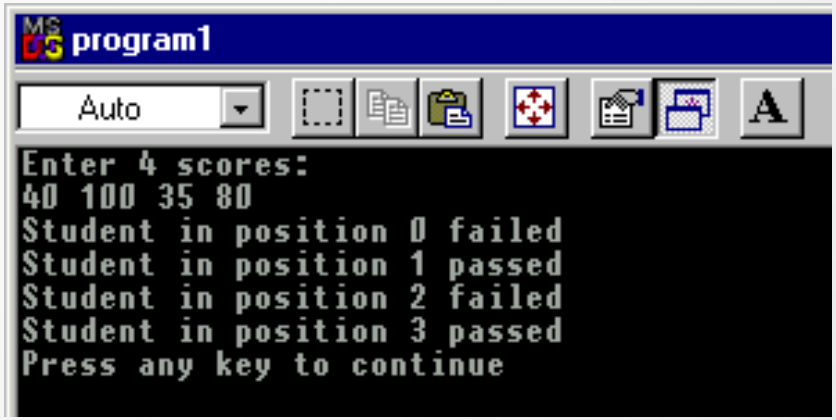
- To step through all indexed variables of an array, use a for statement:

```
for (int index = 0;
     index < Declared_size_of_Array;      index++)
{
    Do something to your_array[index]
}
```

Using Arrays

- Example

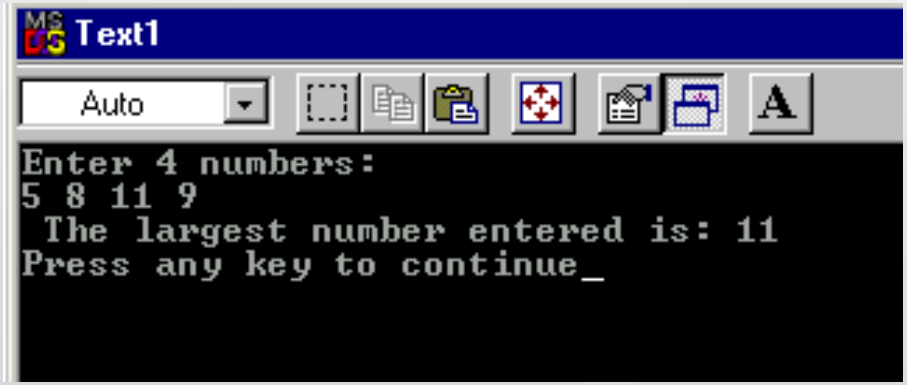
```
#include <iostream>
int main()
{     using namespace std;
      int i, score[4];
      cout<< "Enter 4 scores:\n";
      for (i=0; i<4; i++)
      {     cin >> score[i];
            if (score[i] >= 60)
            cout << "Student in position " << i <<" passed\n";
            else
            cout << "Student in position " << i <<" failed\n";
      }
}
```



Using Arrays

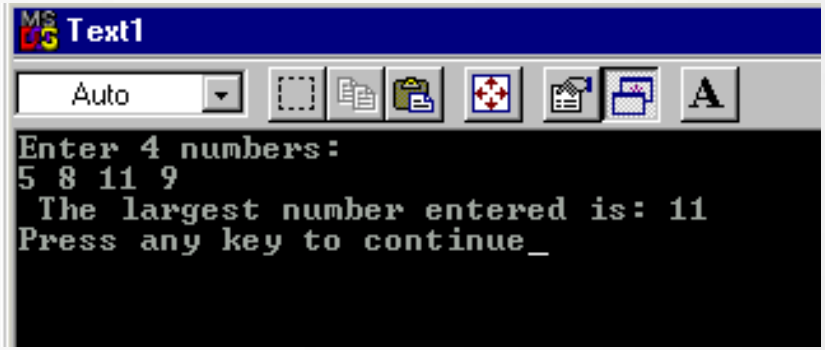
- Try This:

  Write a program that reads 4 positive integers from the user, stores them into an array then finds the maximum number



Using Arrays

- Sample Solution:




```
#include <iostream>
int main()
{     using namespace std;
      int i, my_list[4], max=-1;
      cout<< "Enter 4 numbers:\n";
      for (i=0; i<4; i++)
      {
            cin >> my_list[i];
            if (my_list[i] >= max)
            max = my_list[i];
      }
      cout<< " The largest number entered is: " <<max <<endl;
}
```

Arrays in Functions

- You can use indexed variables or entire arrays as arguments to functions
- Indexed Variables
  - my_function(array[3]);
- Entire Arrays
  - get_scores(score, number_of_scores);

Arrays in Functions

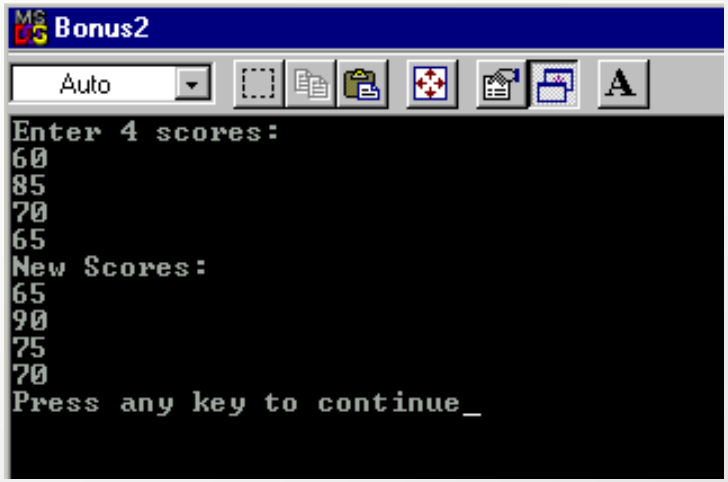
- Indexed Variables

```
include<iostream>
int give_bonus(int old_score);

main()
{     using namespace std;
      int score[4], number;

      cout << "Enter 4 scores: \n";
      for (number = 0; number < 4; number++)
      cin>> score[number];
      for (number = 0; number < 4; number++)
      score[number] = give_bonus(score[number]);
      cout<< "New Scores: \n";
      for (number = 0; number < 4; number++)
      cout<< score[number]<< endl;;}

int give_bonus(int old_score)
{return (old_score+5);}
```



Arrays in Functions

- Entire Arrays

Syntax:

Type_returned Function_name (.., BaseType Array_Name[], …);

- Example

void sum_array (double a[], int size);

→ call: sum_array (a, 5)

Include the size because only the address of the 1st element is passed

- Using *Const*

void sum_array (const double a[], int size);

Searching Partially Filled Arrays

- Example:

```
"F:\C++\May19\Debug\Search.exe"
Enter up to 20 nonnegative whole numbers.
 Mark the end of the list with a negative number.
5
6
-9
Enter a number to search for: 5
5 is in position 0
Search again? _
```

```
#include<iostream>
const int Declared_Size=20;

void fill_array(int a[], int size, int& number_used);

int search(const int a[], int number_used, int target);
```

Searching Arrays Example…continued

```
int main()
{     using namespace std;
      int arr[Declared_Size], list_size, target;
      fill_array(arr, Declared_Size, list_size);
      char ans;
      int result;
      do
      {
        cout<<"Enter a number to search for: ";
        cin>> target;
        result = search(arr, list_size, target);
        if (result==-1)
                cout<< target<< " is not on the list\n";
        else
                cout<< target << " is in position " << result <<endl;
        cout<< "Search again? ";
        cin>>ans;
      } while((ans!='n')&&(ans!='N'));
      cout <<"End of program\n";
      }
```

Searching Arrays Example…continued

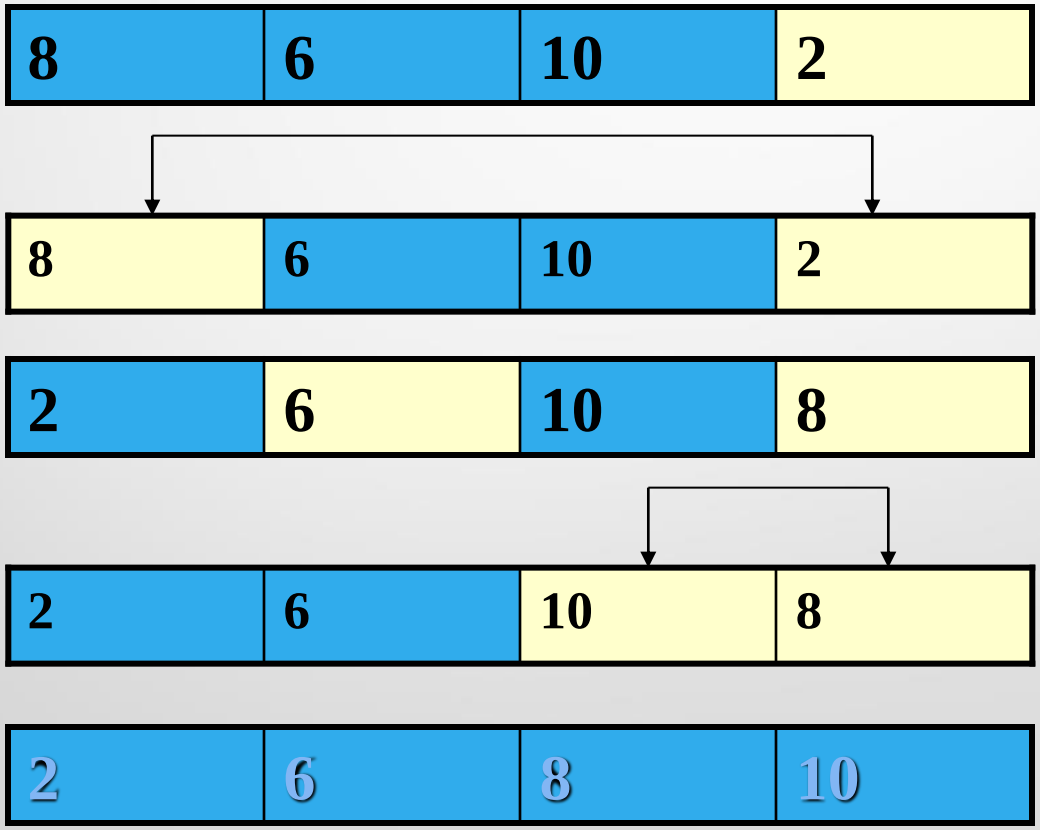
```
int main()
{     using namespace std;
      int arr[Declared_Size], list_size, target;
      fill_array(arr, Declared_Size, list_size);
      char ans;
      int result;
      do
      {
        cout<<"Enter a number to search for: ";
        cin>> target;
        result = search(arr, list_size, target);
        if (result==-1)
                cout<< target<< " is not on the list\n";
        else
                cout<< target << " is in position " << result <<endl;
        cout<< "Search again? ";
        cin>>ans;
      } while((ans!='n')&&(ans!='N'));
      cout <<"End of program\n";
      }
```

Searching Arrays Example…continued

```
int search(const int a[], int number_used, int target)
{
    int index =0;
    bool found = false;
    while ((!found)&& (index<number_used))
    if (target==a[index])
      found = true;
    else
      index++;
    if (found)    return index;
    else       return -1;
}
```

Sorting Arrays

- Selection Sort



Sorting Arrays

- Example: A program that sorts an array of up to 10 positive numbers

```
"F:\C++\May19\Debug\Sort.exe"
This program sorts numbers from lowest to highest.
Enter up to 10 nonnegative whole numbers.
Mark the end of the list with a negative number.
8 6 10 2
-1
In sorted order the numbers are:
2 6 8 10
Press any key to continue
```

Sorting Arrays

- Example:

```
#include<iostream>

void fill_array(int a[], int size, int& number_used);
void sort(int a[], int number_used);
void swap_values (int& v1, int& v2);
int index_of_smallest(const int a[], int start_index, int number_used);
int main()
{     using namespace std;
      cout<<"This program sorts numbers from lowest to highest.\n";

      int sample_array[10], number_used;
      fill_array(sample_array, 10, number_used);
      sort (sample_array, number_used);

      cout<<"In sorted order the numbers are:\n";
      for (int index=0; index<number_used; index++)
        cout<<sample_array[index] << " "; cout<<endl;
}
```

Sorting Arrays

- Example (continued):

```
Void sort(int a[], int number_used)
{
    int index_of_next_smallest;
    for (int index = 0; index<number_used-1; index++)
    {
     index_of_next_smallest= index_of_smallest(a, index, number_used);
     swap_values(a[index], a[index_of_next_smallest]);
    }
}
```

Sorting Arrays

- Example (continued):

```
void swap_values (int& v1, int& v2)
{   int temp;
    temp = v1;
    v1=v2;
    v2=temp;}

int index_of_smallest (const int a[], int start_index, int number_used)
{   int min= a[start_index],
    index_of_min = start_index;
    for (int index=start_index+1; index<number_used; index++)
    if (a[index]<min)
    {       min = a[index];
      index_of_min = index;
    }
    return index_of_min;
}
```

Arrays and Classes

- Array of Structs or Classes

```
class Student
{public:
    int id;
    char Name[10];
    char Major[2];
};
```

Index	ID	Name	Major
0	999	Sara	CS
1	555	Nora	IS
2	222	May	IS

- Student
  - ID
  - Name
  - Major

- Student School[500];

Arrays and Classes

```
Student School[500];
…
For (int i=0; i<500; i++)
{ cout<<“Enter id: “;cin>>School[i].id;
    cout<<“Name: “;     cin>>School[i].Name;
    cout<<“Major: “;  cin>>School[i].Major;
}
```

Arrays and Classes

- Class with array members

```
class Student
{
 public:
    int ID;
    int Grades[10];
}
```

Arrays and Classes

- Try this:

Write a program that reads data into an array of 5 items where each item is a class such as the following:

- Item
  - Code
  - Price

Index	Code	Price
0	123	5.25
1	456	6.30
2	789	9.50

Make *item* an ADT:

- Use member functions to read/write data as well as constructors in your class definition.
- Code &Price should be private members.

Thank You