Assignment-2

UCS540 (Data Structures and Algorithms)

Submitted by Harpartap Singh, 102104119

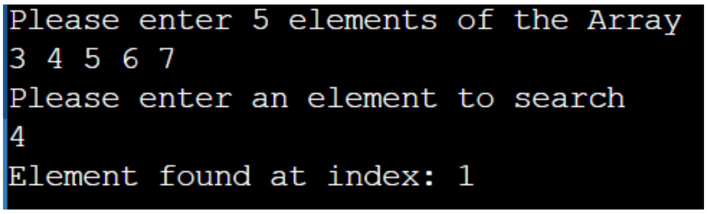
Group- 3EE3

1. Write a program to check whether a given number is present in an array or not (Linear search).

Code:

```
#include<iostream>
using namespace std;
void linearSearch(int a[], int n) {int temp = -1;
for (int i = 0; i < 5; i++) {
if (a[i] == n) {cout << "Element found at position: " << i + 1 << endl;
temp = 0;
break;}}
if (temp == -1) {cout << "No Element Found" << endl;}}
int main() {int arr[5];
cout << "Please enter 5 elements of the Array" << endl;
for (int i = 0; i < 5; i++) {cin >> arr[i];}
cout << "Please enter an element to search" << endl;
int num;
cin >> num;
linearSearch(arr, num);}
```

Output:



```
Please enter 5 elements of the Array
3 4 5 6 7
Please enter an element to search
4
Element found at index: 1
```

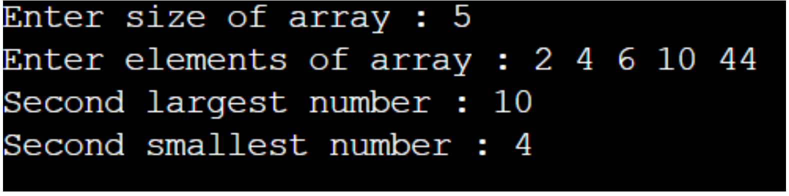
---

2. Write a program to get second maximum and second minimum elements in an array.

Code:

```
#include<iostream>
using namespace std;
int main (){int A[10], n, i, j, x;
cout << "Enter size of array : ";
cin >> n;
cout << "Enter elements of array : ";
for (i = 0; i < n; i++){cin >> A[i];
for (i = 0; i < n; i++){for (j = i + 1; j < n; j++){if (A[i] < A[j]) {x = A[i];
A[i] = A[j];
A[j] = x;}}}
cout << "Second largest number : " << A[1];
cout << "\nSecond smallest number : " << A[n - 2];}
```

Output:



```
Enter size of array : 5
Enter elements of array : 2 4 6 10 44
Second largest number : 10
Second smallest number : 4
```

---

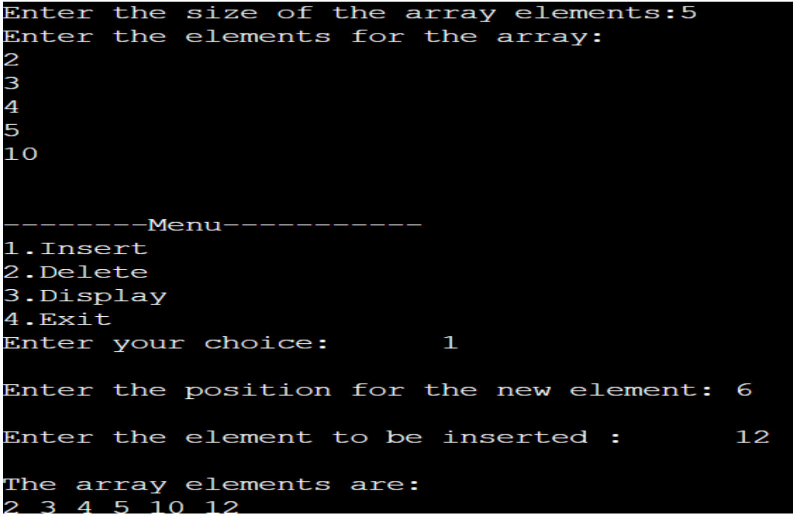
3. Write a program to perform insertion (any location), deletion (any location) and traversal in an array.

Code:

```
#include<iostream>
using namespace std;
int a[20],b[20],c[40];
int m,n,p,val,i,j,key,pos,temp;
void display();
void insert();
void del();
int main(){int choice;
cout<<"Enter the size of the array elements:";
cin>>n;
cout<<"Enter the elements for the array:\n";
for (i=0;i<n;i++){cin>>a[i];}
do {cout<<"\n\n--------Menu-----------\n";
cout<<"1.Insert\n";
cout<<"2.Delete\n";
cout<<"3.Display\n";
cout<<"4.Exit";
cout<<"\nEnter your choice:\t";
cin>>choice;
switch (choice){
case 1: insert();
break;
case 2: del();
break;
case 3:display();
break;
default :cout<<"\nProgram Ends:\n";}} while (choice!=4);
return 0;}
void display(){int i;
cout<<"\nThe array elements are:\n";
for(i=0;i<n;i++){
cout<<a[i]<<" ";}}
```

```
void insert(){cout<<"\nEnter the position for the new element:\t";
cin>>pos;
cout<<"\nEnter the element to be inserted :\t";
cin>>val;
for (i=n; i>=pos-1; i--){a[i+1]=a[i];}
a[pos-1]=val;
n=n+1;
display();}
void del(){
cout<<"\n Enter the position of the element to be deleted:\t";
cin>> pos;
val= a [pos];
for (i= pos;i<n-1;i++){a[i]=a[i+1];}
n=n-1;
cout<<"\nThe deleted element is = "<<val;
display();}
```

Output:



```
Enter the size of the array elements:5
Enter the elements for the array:
2
3
4
5
10


--------Menu-----------
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:      1

Enter the position for the new element: 6

Enter the element to be inserted :      12

The array elements are:
2 3 4 5 10 12
```

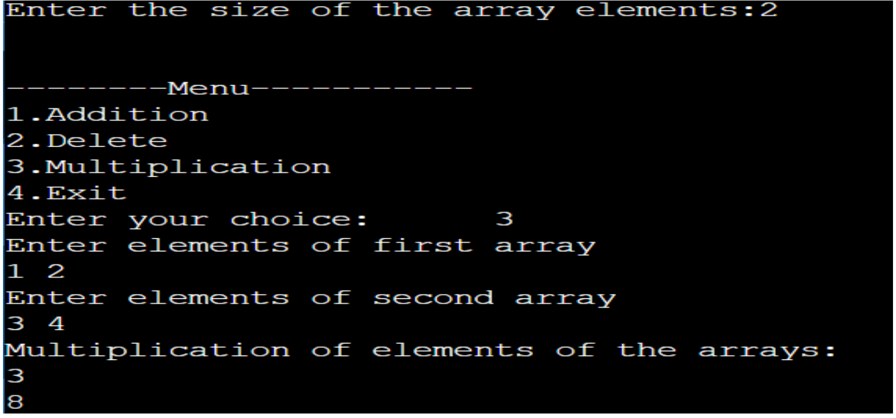
---

4. Write a menu driven program to perform addition, multiplication and subtraction of 2 arrays.

Code:

```
#include<iostream>
using namespace std;
int a[20],b[20],s[20],c,n,d[20],m[20];
void add();
void sub();
void multiply();
int main()
{int choice;
cout<<"Enter the size of the array elements:";
cin>>n;
do {cout<<"\n\n--------Menu-----------\n";
cout<<"1.Addition\n";
cout<<"2.Delete\n";
cout<<"3.Multiplication\n";
cout<<"4.Exit";
cout<<"\nEnter your choice:\t";
cin>>choice;
switch (choice){case 1: add();
break;
case 2: sub();
break;
case 3:multiply();
break;
default :cout<<"\nProgram Ends\n"}} while (choice!=4);
return 0;}
void add(){cout << "Enter elements of first array" << endl;
for (c = 0; c < n; c++){cin >> a[c];}
cout << "Enter elements of second array" << endl;
for (c = 0; c < n; c++){cin >> b[c];}
cout << "Sum of elements of the arrays:" << endl;
for (c = 0; c < n; c++) {s[c] = a[c] + b[c];
cout << s[c]<<endl;}}
void sub(){cout << "Enter elements of first array" << endl;
for (c = 0; c < n; c++){cin >> a[c];}
cout << "Enter elements of second array" << endl;
for (c = 0; c < n; c++){cin >> b[c];}
cout << "Difference of elements of the arrays:" << endl;
for (c = 0; c < n; c++) {d[c] = a[c]-b[c];
cout << d[c]<<endl;}}
void multiply(){cout << "Enter elements of first array" << endl;
for (c = 0; c < n; c++){cin >> a[c];}
cout << "Enter elements of second array" << endl;
for (c = 0; c < n; c++){cin >> b[c];}
cout << "Multiplication of elements of the arrays:" << endl;
for (c = 0; c < n; c++) {m[c] = a[c]*b[c];
cout << m[c]<<endl;}}
```

Output:



```
Enter the size of the array elements:2


--------Menu------------
1.Addition
2.Delete
3.Multiplication
4.Exit
Enter your choice:      3
Enter elements of first array
1 2
Enter elements of second array
3 4
Multiplication of elements of the arrays:
3
8
```

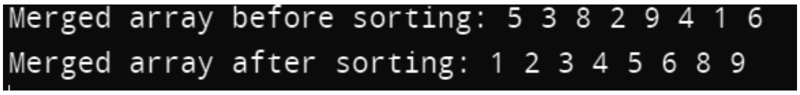
5. Write a program to perform sorting while merging (Merge two sorted arrays into one sorted array).

Code:

```
#include <iostream>
```

```
using namespace std;
void mergeArrays(int arr1[], int size1, int arr2[], int size2, int merged[]){
for (int i = 0; i < size1; i++) {merged[i] = arr1[i];}
for (int i = 0; i < size2; i++) {merged[size1 + i] = arr2[i];}}
void insertionSort(int arr[], int size){for (int i = 1; i < size; i++){int key = arr[i];
int j = i – 1;
while (j >= 0 && arr[j] > key){arr[j + 1] = arr[j];
j = j – 1;}
arr[j + 1] = key;}}
void printArray(int arr[], int size){for (int i = 0; i < size; i++){std::cout << arr[i] << "";}
std::cout << std::endl;}
int main(){int arr1[] = {5, 3, 8, 2};
int size1 = sizeof(arr1) / sizeof(arr1[0]);
int arr2[] = {9, 4, 1, 6};
int size2 = sizeof(arr2) / sizeof(arr2[0]);
int merged[size1 + size2];
mergeArrays(arr1, size1, arr2, size2, merged);
std::cout << "Merged array before sorting: ";
printArray(merged, size1 + size2);
insertionSort(merged, size1 + size2);
std::cout << "Merged array after sorting: ";
printArray(merged, size1 + size2);}
```

Output:



```
Merged array before sorting: 5 3 8 2 9 4 1 6
Merged array after sorting: 1 2 3 4 5 6 8 9
```

---

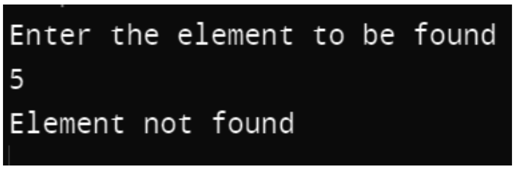
6. Write the above programs (1,2, and 3) using functions and call by address only.

Code:

(a)

```
#include <iostream>
Using namespace std;
int linearSearch(int *arr, int size, int key) {for (int i = 0; i < size; ++i) {
if (*arr == key) {return i; // Return index if key found}
++arr;}
return -1; // Return -1 if key not found}
int main() {int arr[] = {10, 20, 30, 40, 50};
int size = sizeof(arr) / sizeof(arr[0]);
int key;
std::cout<<"Enter the element to be found"<<std::endl;
std::cin>>key;
int index = linearSearch(arr, size, key);
if (index != -1) {std::cout << "Element found at index: " << index << std::endl;} else {
std::cout << "Element not found" << std::endl;}}
```

Output:

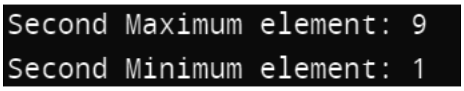


```
Enter the element to be found
5
Element not found
```

(b)Code:

```
#include <iostream>
Using namespace std;
void findSecondMinMax(int *arr, int size, int &secondMax, int &secondMin){secondMax = arr[0];
secondMin = arr[0];
for (int i = 1; i < size; ++i) {if (arr[i] > secondMax){secondMax = arr[i];}
else if (arr[i] < secondMin){secondMin = arr[i];}
int max = arr[0];
int min = arr[0];
for (int i = 0; i < size; ++i){if (arr[i] != secondMax && arr[i] > max){
secondMax = arr[i];}
if (arr[i] != secondMin && arr[i] < min){secondMin = arr[i];}}}
int main(){int arr[] = {5, 3, 8, 2, 1, 9};
int size = sizeof(arr) / sizeof(arr[0]);
int secondMax, secondMin;
findSecondMinMax(arr, size, secondMax, secondMin);
std::cout << "Second Maximum element: " << secondMax << std::endl;
std::cout << "Second Minimum element: " << secondMin << std::endl;}
```

Output:



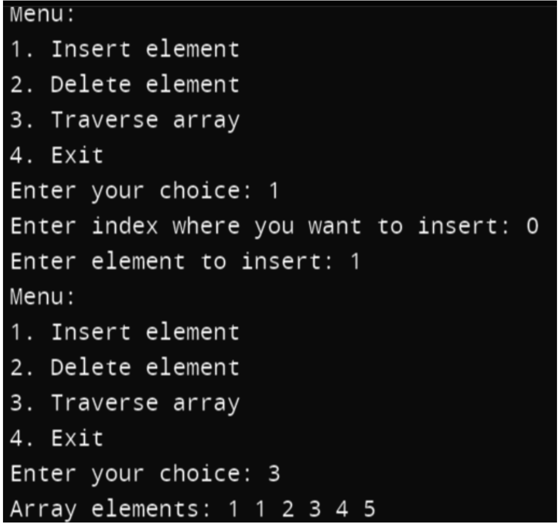
```
Second Maximum element: 9
Second Minimum element: 1
```

(c)

Code:

```
#include <iostream>
Using namespace std;
void insertElement(int *arr, int &size, int index, int element);
void deleteElement(int *arr, int &size, int index);
void traverseArray(int *arr, int size);
void displayMenu();
int main(){ const int MAX_SIZE = 10;
int arr[MAX_SIZE] = {1, 2, 3, 4, 5};
int size = 5;
int choice;
do{displayMenu();
std::cin >> choice;
switch (choice) {case 1: {int index, element;
std::cout << "Enter index where you want to insert: ";
std::cin >> index;
std::cout << "Enter element to insert: ";
std::cin >> element;
insertElement(arr, size, index, element);
break;}
case 2: {int index;
std::cout << "Enter index from where you want to delete: ";
std::cin >> index;
deleteElement(arr, size, index);
break;}
case 3: {traverseArray(arr, size);
break;}
case 4:
std::cout << "Exiting program." << std::endl;
break;
default: std::cout << "Invalid choice. Please try again." << std::endl;}}
while (choice != 4);}
void insertElement(int *arr, int &size, int index, int element){
if (index < 0 || index > size){std::cout << "Invalid index for insertion." << std::endl;
return;}
for (int i = size; i > index; --i){arr[i] = arr[i - 1];}
arr[index] = element;
size++;}
void deleteElement(int *arr, int &size, int index){if (index < 0 || index >= size){std::cout << "Invalid index for deletion." << std::endl;
return;}
for (int i = index; i < size - 1; ++i){arr[i] = arr[i + 1];}
size--;}
void traverseArray(int *arr, int size) {std::cout << "Array elements: ";
for (int i = 0; i < size; ++i){std::cout << arr[i] << " ";}
std::cout << std::endl;}
void displayMenu(){std::cout << "\nMenu:\n";
std::cout << "1. Insert element\n";
std::cout << "2. Delete element\n";
std::cout << "3. Traverse array\n";
std::cout << "4. Exit\n";
std::cout << "Enter your choice: ";}
```

Output:



```
Menu:
1. Insert element
2. Delete element
3. Traverse array
4. Exit
Enter your choice: 1
Enter index where you want to insert: 0
Enter element to insert: 1
Menu:
1. Insert element
2. Delete element
3. Traverse array
4. Exit
Enter your choice: 3
Array elements: 1 1 2 3 4 5
```