

15B17CI371 – Data Structures Lab

ODD 2024
Week 3-LAB A
Practice Lab
[CO: C270.2]

1)

```
#include<iostream>

using namespace std;

void firstRepeatingPair (int a[], int n)

{

    bool check=false;

    for (int i=0;i<n-1;i++){

        for (int j=i+1;j<n-1;j++){

            if(a[i]==a[j]&& a[i+1]==a[j+1]){

                check=true;

                cout<<"Pair: \t"<<a[i]<<" "<<a[i+1]<<endl;

                cout<<"Output:\t"<<j+1<<endl;

                break;

            }

        }

    }

    if(!check) {cout<<"No repeating pairs found. "<<endl; }

}

int main()

{
```

```

    int a,n;

    cout<<"Enter Number of elements in array:\n";

    cin>>n;

    int *arr=new int[n];

    cout<<"\nEnter elements in array:\n";

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

        cin>>a;

        arr[i]=a;

    }

    firstRepeatingPair (arr, n);

    return 0;

}

```

```

Enter Number of elements in array:
6

Enter elements in array:
1
2
3
4
1
2
Pair: 1,2
Output: 5
kavyamalik@Kavyas-MacBook-Air sem3.c %

```

2)

```

#include<iostream>

#include<cmath>

using namespace std;

int closestSum (int a[], int N, int sum)

{

int left=0, right=N-1;

int csum=a[left]+a[right];

int lsum=a[left], rsum=a[right];

while (left<right)

{

int curr_sum=a[left]+a[right];

if (abs (curr_sum) <abs (csum))

{

csum=curr_sum;

lsum=a[left];

rsum=a[right];

}

if (curr_sum<sum) {left++; }

else{right--;}

}

return csum;

}

```

```

int main(){

int N, sum;

cout<<"Enter no. of elements (N):";

cin>>N;

int arr[N];

cout<<endl<<"Enter all the "<<N<<" elements:"<<endl<<endl;

for (int i=0;i<N; i++) { cin>>arr[i];}

cout<<endl<<"Enter a closest sum value:";

cin>>sum;

for (int i=0;i<N-1;i++)

{

for (int j=0;j<N-i-1;j++)

{

if (arr[j]>arr[j+1])

{

int temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

}

}

}

cout<<endl<<"Output: "<<closestSum (arr, N, sum) <<endl;

}

```

```
Enter no. of elements (N):4
```

```
Enter all the 4 elements:
```

```
1  
2  
3  
4
```

```
Enter a closest sum value:0
```

```
Output: 3
```

```
kavyamalik@Kavyas-MacBook-Air sem3.c %
```

3)

```
#include<iostream>
```

```
using namespace std;
```

```
void missingAP (int arr[], int N)
```

```
{
```

```
int d =(arr[N-1]-arr[0])/N;
```

```
for (int i=1;i<N;i++)
```

```
{
```

```
if (arr[i]!=arr[0]+i*d){
```

```
cout<<"The missing AP term is "<<arr[0]+i*d<<endl;
```

```
return;
```

```
}
```

```
}
```

```

cout<<"No AP term missing"<<endl;

}

int main()

{

int A[6]={2,4,6,10,12,14};

cout<<"A={";

for (int i=0;i<6;i++)

{

if (i<5)

{cout<<A[i]<<" ";

}

else{ cout<<A[i]<<"}"<<endl<<endl; }

}

missingAP (A, 6);

return 0;

}

```

```
A={2,4,6,10,12,14}
```

```
The missing AP term is 8
```

```
kavyamalik@Kavyas-MacBook-Air sem3.c %
```

4)

```

#include<iostream>

using namespace std;

void findFirstAndLastOccurrence (int A[], int N, int x, int &f, int &l)
{
    int low=0, high=N-1, first=-1, last=-1;

    while (low<=high)
    {
        int mid=low+ (high-low)/2;

        if (A[mid] ==x)
        {

            if (first== -1 || mid<first)
            {first=mid;}

            if (mid>last) {last=mid; }

            int temp=mid;

            while (--temp>=low && A[temp]==x) {f=temp; }

            temp=mid;

            while (++temp<=high&&A[temp]==x) {l=temp; }

            break;

        }

        else if (A[mid]>x) {high=mid-1;}

        else{low=mid+1;}

    }

}

int main()

{

    int A[]={1,2,2,2,3,4,5};

```

```

cout<<"A[]={";
for (int i=0;i<7;i++)
{
if (i<6)
{
    cout<<A[i]<<" ";
}
else{ cout<<A[i]<<"}"<<endl<<endl; }
}
int x=2;
cout<<"x="<<x<<endl<<endl;
int first, last;
findFirstAndLastOccurrence (A, 6, x, first, last);
if (first!=-1&&last!=-1)
{
    cout<<"First occurrence of "<<x<<" is at "<<first+1<<"th postion"<<endl;
    cout<<"Last occurrence of "<<x<<" is at "<<last+1<<"th postion"<<endl;
}
else { cout<<x<<"Repeating occurence not found"<<endl; }
return 0;
}

```

```
A[]={1,2,2,2,3,4,5}
```

```
x=2
```

```
First occurrence of 2 is at 2th postion
Last occurrence of 2 is at 4th postion
```


5)

```
#include<iostream>
using namespace std;
void interpolSearch (int a[], int N, int K)
{
    int low=0, high=N-1;
    while (low<=high && K>=a[low] && K<=a [high])
    {
        int pos=low+ ((K- a[low])* (high-low))/(a[high]-a[low]);
        if (a[pos]==K)
        {
            cout<<"K="<<K<<" found at "<<pos+1<<"th postion"<<endl;
            return;
        }
        if (a[pos] <K) {low=pos+1;}
        else {high=pos-1; }
    }
    cout<<"K="<<K<<" not found in the array"<<endl;
}
int main()
{
    int arr[]={10,12,13,16,18,19,20,21,22,23};
    cout<<"A[]={";
    for (int i=0;i<10;i++)
    {
        if (i<9) {cout<<arr[i]<<",";}
        else{ cout<<arr[i]<<"}"<<endl<<endl; }
    }
    int K=20;
    interpolSearch (arr, 10, K);
    return 0;
}
```

A[]={10,12,13,16,18,19,20,21,22,23}

K=20 found at 7th postion

6)

```
#include<iostream>

using namespace std;

bool checkSubArr(int a[], int n, int len)

{
```

```

for (int i=0;i<=n-len;i++){
    if(a[i]>=a[i+len-1]) {return true; }
}
return false;
}

int maxSubLength (int arr[], int n){
    int low=2, high=n, result=1;
    while (low<=high)
    {
        int mid=low+(high-low)/2;
        if (checkSubArr (arr, n, mid))
        {
            result=mid;
            low=mid+1;
        }
        else
        {
            high=mid-1;
        }
    }
    return result;
}

int main()
{
    int n;

    cout<<"Enter the no. of elements (n): \t";

    cin>>n;

    int arr[n];

```

```

cout<<endl<<"Enter all the "<<n<<" elements: "<<endl<<endl;

for (int i=0;i<n;i++) { cin>>arr[i];}

cout<<endl;

cout<< "The maximum length subarray where the 1st element is greater than or equal to the
last element is: " <<maxSubLength (arr, n) << endl;

return 0;

}

```

```

Enter the no. of elements (n): 6
Enter all the 6 elements:
-5 -1 7 5 1 -2
The maximum length subarray where the 1st element is greater than or equal to the last element is: 5

```

7)

```

#include<iostream>

#include<cstring>

using namespace std;

int strIndex (string a[], int N, string x){
for (int i=0;i<N;i++){
if(a[i]==x)
{return i;}
}
return -1;
}

int main()
{
string s[]{"Hi", "Folks", "ide", "for", "practice"};

cout<<"arr[]={";

```

```

for (int i=0;i<5;i++)
{
if (i<4)
{
    cout<<"\ "<<s[i]<<" \", ";
}
else
{ cout<<" \ " <<s[i]<<" \}"<<endl<<endl; }
}

string x;

cout<<"Enter a string (x):\t";

getline (cin, x);

cout<<"Output: \t"<<strIndex (s, 5, x) <<endl;

return 0;
}

```

```
arr[]={"Hi ", "Folks ", "ide ", "for ", " practice "}
```

```
Enter a string (x):    ide
```

```
Output:              2
```

```
arr[]={"Hi ", "Folks ", "ide ", "for ", " practice "}
```

```
Enter a string (x):    zz
```

```
Output:              -1
```

8)

```
#include<iostream>
```

```
using namespace std;
```

```
bool linearSearch (int a[], int N, int K)
```

```

{
int pos;
for (pos=0;pos<N; pos++)
{
if(a[pos]==K)
{return true; }
}
return false;
}

int main(){
int arr[]={10,12,13,16,18,19,20,21,22,23};
cout<<"arr[]={";
for (int i=0;i<10;i++)
{
if (i<9)
{cout<<arr[i]<<" ";}
else
{cout<<arr[i] <<"}"<<endl<<endl; }
}
int K;
cout<<"Enter an element to be searched (K):\t";
cin>>K;
if (linearSearch (arr, 10, K))
{cout<<"K="<<K<<" present in the array"<<endl; }
else {cout<<"K="<<K<<" absent in the array"<<endl; }
return 0;
}

```

```
arr[]={10,12,13,16,18,19,20,21,22,23}
```

```
Enter an element to be searched (K): 21
```

```
K=21 present in the array
```

```
arr[]={10,12,13,16,18,19,20,21,22,23}
```

```
Enter an element to be searched (K): 25
```

```
K=25 absent in the array
```