Indian Institute of Technology (Indian School of Mines), Dhanbad

Algorithm Design & Analysis

Submitted to:	Submitted by:
Prof. Saumen Bag	Shubham Maurya 16JE002437 4th Sem., CSE

1.Let R and S are two relations on a given set of positive integers A with set cardinality N.

Define a proper data structure to represent R and S. Now, perform the following operations as given below.

I. R U S

ii. R ^ S

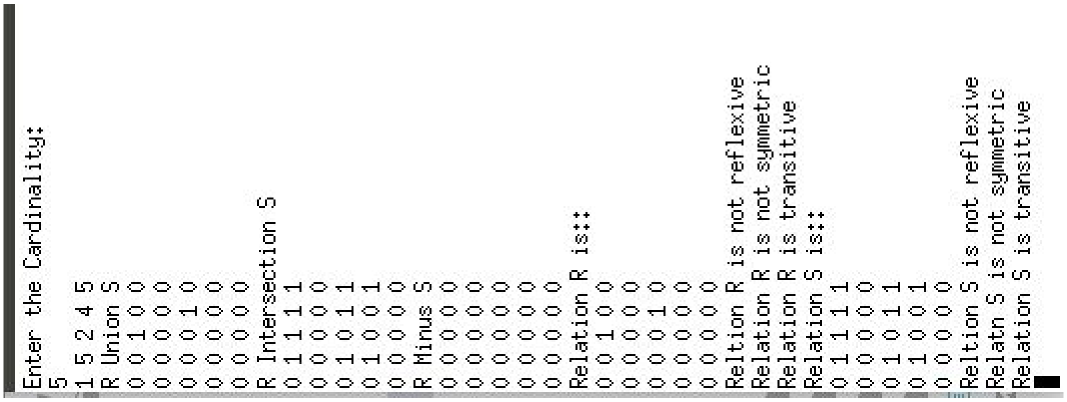
iii. R-S

iv. Determine whether the relation R or S is (a) reflexive and/or irreflexive, (b) symmetric and /or antisymmetric, and (c) transitive.

```
#include <iostream>
using namespace std;
int main(){
        int n,temp=0;
        cout << "Enter the Cardinality:\n";
        cin >> n;
        int a[n],i,j;
        for(i=0;i<n;i++)
                cin >> a[i];
        int R[n][n] , S[n][n];
        for(i=0;i<n;i++){
                for(j=0;j<n;j++){
                        R[i][j] = 0;
                        S[i][j] = 0;
       }}
        for(i=0;i<n;i++){
                for(j=0;j<n;j++){
                        if(a[j]== 2*a[i])
                                R[i][j] = 1;
                        if(a[j] > a[i])
                                S[i][j] = 1;}}
        int RUS[n][n],RIS[n][n],RMS[n][n];
        for(i=0;i<n;i++){
                for(j=0;j<n;j++){
                        RUS[i][j] = R[i][j] & S[i][j];
                        RIS[i][j] = R[i][j] | S[i][j];
        if(R[i][j]==1 && S[i][j]==1)
            RMS[i][j] = 0;
        else
            RMS[i][j] = R[i][j];  }}
cout << "R Union S\n";
for(i=0;i<n;i++){
    for(j=0;j<n;j++)
        cout << RUS[i][j] << " ";
    cout << "\n";
}
cout << "R Intersection S\n";
for(i=0;i<n;i++){
    for(j=0;j<n;j++)
        cout << RIS[i][j] << " ";
    cout << "\n";
}
cout << "R Minus S\n";
for(i=0;i<n;i++){
    for(j=0;j<n;j++)
        cout << RMS[i][j] << " ";
    cout << "\n";
}
cout<<"Relation R is::\n";
for(i=0;i<n;i++){
    for(j=0;j<n;j++)
        cout<<R[i][j]<<" ";
    cout<<"\n";
}
for(int i=1;i<=n;i++){
    if(R[i][i]==0){
        temp++;
        break;}}
if(temp>0)
    cout<<"Reltion R is not reflexive\n";
else
    cout<<"Reltion R is reflexive\n";
temp=0;

for(int i=0;i<=n;i++){
    for(int j=1;j<=n;j++){
        if(R[i][j]==1){
            if(R[j][i]==0){
                temp++;
```

```
                                        break;
}}}}
if(temp>0)
          cout<<"Relation R is not symmetric\n";
else
          cout<<"Relation R is symmetric\n";
temp=0;
for(int i=0;i<=n;i++){
          for(int j=1;j<=n;j++){
                    if(R[i][j]==1){
                              for(int k=1;k<=n;k++){
                                        if(R[j][k]=1){
                                                  if(R[i][k]=0){
                                                            temp++;
                                                            break;
}}}}}}
if(temp>0)
          cout<<"Relation R is not transitive\n";
else
          cout<<"Relation R is transitive\n";
temp=0;
cout<<"Relation S is::\n";
for(i=0;i<n;i++){
          for(j=0;j<n;j++)
                    cout<<S[i][j]<<" ";
          cout<<"\n";
}
for(int i=1;i<=n;i++){
          if(S[i][i]==0){
                    temp++;
                    Break;  }}
if(temp>0)
          cout<<"Reltion S is not reflexive\n";
else
          cout<<"Reltion S is reflexive\n";
temp=0;
for(int i=0;i<=n;i++){
          for(int j=1;j<=n;j++){
                    if(S[i][j]==1){
                              if(S[j][i]==0){
                                        temp++;
                                        break;
}}}}
```

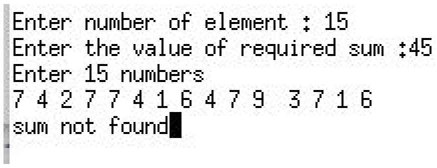
```
	if(temp>0)
		cout<<"Relatn S is not symmetric\n";
	else
		cout<<"Relatn S is symmetric\n";
	temp=0;
	for(int i=0;i<=n;i++){
		for(int j=1;j<=n;j++){
			if(S[i][j]==1){
				for(int k=1;k<=n;k++){
					if(S[j][k]=1){
						if(S[i][k]=0){
							temp++;
							break;
	}}}}}}
	if(temp>0)
		cout<<"Relation S is not transitive\n";
	else
		cout<<"Relation S is transitive\n";

	return 0;
}
```



```
Enter the Cardinality:
5
1 5 2 4 5
R Union S
0 0 1 0 0
0 0 0 0 0
0 0 0 1 0
0 0 0 0 0
0 0 0 0 0
R Intersection S
0 1 1 1 1
0 0 0 0 0
0 1 0 1 1
0 1 0 0 1
0 0 0 0 0
R Minus S
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
Relation R is::
0 0 1 0 0
0 0 0 0 0
0 0 0 1 0
0 0 0 0 0
0 0 0 0 0
Reltion R is not reflexive
Relation R is not symmetric
Relation R is transitive
Relation S is::
0 1 1 1 1
0 0 0 0 0
0 1 0 1 1
0 1 0 0 1
0 0 0 0 0
Reltion S is not reflexive
Relatn S is not symmetric
Relation S is transitive
```

2.**You are given an array A[ ] of n positive integers, and a target sum t (again a positive integer). Your task is to find a non-empty sub-array A[i...j]such that t = A[i] + A[i+1] + ... + A[j], or report that no such sub-array exist**s.

```
#include<bits/stdc++.h>
using namespace std;
int mem[100];
int a[2000];

int dp(int n,int t){
    if(n==0 && t==0)
        return 0;
    if(t!=0 && n==0)
        return 0;
    if(t==0)
        return 1;
    if(mem[t]!=-1)
        return mem[t];
    return mem[t]=max(dp(n-1,t),dp(n-1,t-a[n]));
}

int main(){
    int t,n  ;
    cout<<"Enter number of element :";
    cin>>n;
    cout<<"Enter the value of required sum :";
    cin>>t;
    cout<<"Enter "<<n<<" numbers"<<endl;
    for(int i=0;i<n;i++){
        cin>>a[i];
    }
    memset(mem,-1,sizeof(mem));
    int ans=dp(n,t);
    if(ans==1)
        cout<<"sum found";
    else
        cout<<"sum not found";
```



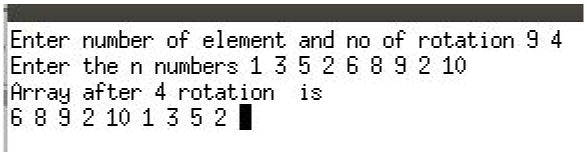
```
Enter number of element : 15
Enter the value of required sum :45
Enter 15 numbers
7 4 2 7 7 4 1 6 4 7 9  3 7 1 6
sum not found
```

```
    return 0;
}
```

3.**You are given an array A of n integers. You are also given an integer s in the range1 ≤ s ≤ n–1. Your task is to right rotate (cyclically right shift) the array A by s cells. Write a program to solve this problem.**

```
#include<bits/stdc++.h>
using namespace std;
int mem[100];
int a[2000];

int gcd(int a,int b){
    if(a%b==0)
        return b;
    if(b%a==0)
        return a;
    if(a>b)
        return gcd(a%b,b);
    if(a<b)
        return gcd(a,b%a);
}

void rotation(int t,int n){
    int l=gcd(t,n);
    for(int i=0;i<l;i++){
        int temp=a[i];
        int j=i;
        while(1){
            int k=j+t;
            if(k>=n)
                k=k-n;
            if(k==i)
                break;
            a[j]=a[k];
            j=k;
        }
        a[j]=temp;
    }
}
```

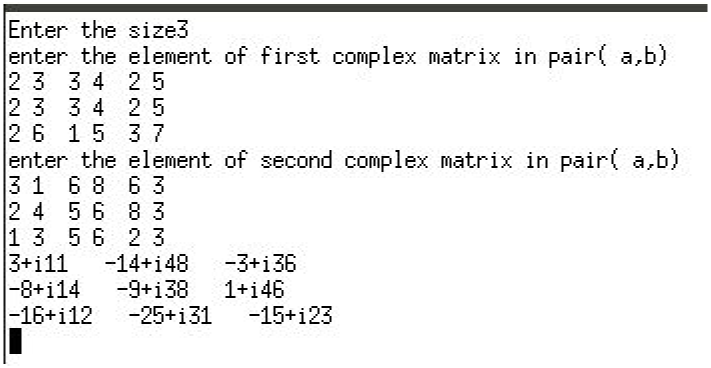
```
int main(){
        int n,t  ;
        cout<<"Enter number of element and no of rotation ";
        cin>>n>>t;
        cout<<"Enter the n numbers ";
        for(int i=0;i<n;i++){
                cin>>a[i];
        }

        rotation(t,n);
        cout<<"Array after "<<t<<" rotation  is "<<endl;
        for(int i=0;i<n;i++)
                cout<<a[i]<<" ";

        return 0;
}
```



```
Enter number of element and no of rotation 9 4
Enter the n numbers 1 3 5 2 6 8 9 2 10
Array after 4 rotation  is
6 8 9 2 10 1 3 5 2 █
```

4.**A complex-valued matrix X is represented by a pair of matrices (M 1R , M 1I), where M 1R And M 1I contain real values. Now, do the following: Define a proper data structure to represent the complex-valued matrix.**

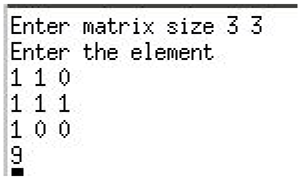
```
#include<iostream>
using namespace std;

struct complex{
        int r,i;
};
int main(){
        int w;
        cout<<"Enter the size";
        cin>>w;
        complex a[w][w],b[w][w],c[w][w];
        cout<<"enter the element of first complex matrix in pair( a,b)"<<endl;
        for(int j=0;j<w;j++){
                for(int k=0;k<w;k++){
                        cin>>a[j][k].r>>a[j][k].i;
                }
        }
        cout<<"enter the element of second complex matrix in pair( a,b)"<<endl;
        for(int l=0;l<w;l++){
                for(int m=0;m<w;m++){
                        cin>>b[l][m].r>>b[l][m].i;
                }
        }

        for(int n=0;n<w;n++){
                for(int t=0;t<w;t++){
                        c[n][t].r=(((a[n][t].r)*(b[n][t].r))-((a[n][t].i)*(b[n][t].i)));
                        c[n][t].i=(((a[n][t].r)*(b[n][t].i))+((a[n][t].i)*(b[n][t].r)));
                }
        }

        for(int s=0;s<w;s++){
                for(int p=0;p<w;p++)
                        cout<<c[s][p].r<<"+i"<<c[s][p].i<<"   ";
                cout<<endl;
        }

        return 0;
}
```



```
Enter the size3
enter the element of first complex matrix in pair( a,b)
2 3  3 4  2 5
2 3  3 4  2 5
2 6  1 5  3 7
enter the element of second complex matrix in pair( a,b)
3 1  6 8  6 3
2 4  5 6  8 3
1 3  5 6  2 3
3+i11   -14+i48   -3+i36
-8+i14   -9+i38   1+i46
-16+i12   -25+i31   -15+i23
```

5.**Suppose that each row of an n × n array A consists of 1′s and 0′s such that, in any row ofA, all the 1′s come before any 0′s in that row. Suppose further that the number of 1′s is at least the number in row i + 1, for i = 0, 1, ... , n − 2. Assuming A is already in memory, write a program running in O(n) time for counting the number of 1′s in the array A**

```
#include<bits/stdc++.h>
using namespace std;
int mem[100][100];
int a[200][200];
int dp(int i,int j){
    if(i==0||j==0)
        return 0;
    if(mem[i][j]!=0)
        return mem[i][j];
    if(a[i][j]==1)
        return mem[i][j]=1+dp(i,j-1);
    if(a[i][j]==0)
        return mem[i][j]=dp(i,j-1);
}
int main(){
    int n,t  ;
    cout<<"Enter matrix size ";
    cin>>n;
    cout<<"Enter the element"<<endl;
    for(int i=1;i<=n;i++){
        for(int j=1;j<=n;j++){
            cin>>a[i][j];
            mem[i][j]=0;
    }}
    mem[n][n]=0;
    for(int i=1;i<=n;i++){
        int ans = dp(i,n);
    }
    int sum=0;
    for(int i=1;i<=n;i++){
        sum=sum+mem[i][n];
    }
    cout<<sum<<endl;
    return 0;
}
```



```
Enter matrix size 3 3
Enter the element
1 1 0
1 1 1
1 0 0
9
```