Contents

Rudra Nil Basu

Summary

1 Pattern

Problem 1.1 *Write a program to print the following pattern using* ***for*** *loop*

```
1
22
333
4444
55555
...
```

Code.

```
#include<stdio.h>
using namespace std;

int main()
{
        int n,i,j;
        scanf("%d",&n);
        for(i=1;i<=n;i++) {
                for(j=1;j<=i;j++) {
                        printf("%d",i);
                }
                printf("\n");
        }
}
```

Output.

```
5
1
22
333
4444
55555
```

2 Average of Cricket Players

Problem 1.2 *A cricket team has the following table of batting figures for a series of test matches*

Player's Name	Runs	Innings	Times not out
Sachin	**8430**	**230**	**180**
Saurav	**4200**	**130**	**9**
Rahul	**3350**	**105**	**11**

Write a program to read figures from the above form, to calculate the batting average and print out the complete table including the average

Code.

```
#include<stdio.h>
#include<vector>
#include<iostream>
using namespace std;

typedef struct stats {
        char name[50];
        int runs;
        int innings;
        int not_out;
        float average;
}stats;

int main()
{
        int i,n;
        char strtr[10];

        int ans;
        while(1) {
                stats players;
                printf("Enter name:");
                scanf("%s",players.name);
                printf("Enter runs, innings, not_out for %s\n",
                players.name);
                scanf("%d %d %d",&players.runs,&players.innings,
                &players.not_out);
                players.average=players.runs*1.0/players.innings;
                g.push_back(players);
                printf("Want more ?(1/0)\nyes=1\tno=0\n");
                scanf("%d",&ans);
                if(ans==0) {
                        break;
                }
        }
        printf("Name\tRuns\tInnings\tNot Out\tAverage\n");
        for(i=0;i<g.size();i++) {
                printf("%s\t%d\t%d\t%d\t%f\n",g[i].name,
                g[i].runs,g[i].innings
                                ,g[i].not_out, g[i].average);
        }
        return 0;
}
```

Output.

```
Enter name:Rahul
Enter runs, innings, not_out for Rahul
```

```
3350 105 11
Want more ?(1/0)
yes=1    no=0
1
Enter name:Sachin
Enter runs, innings, not_out for Sachin
8430
230 18
Want more ?(1/0)
yes=1    no=0
1
Enter name:Saurav
Enter runs, innings, not_out for Saurav
4200 130 9
Want more ?(1/0)
yes=1    no=0
1
Enter name:ThePhenomenalRNB
Enter runs, innings, not_out for ThePhenomenalRNB
8888 105 18
Want more ?(1/0)
yes=1    no=0
0
Name     Runs     Innings Not Out Average
Rahul    3350     105      11      31.904762
Sachin   8430     230      18      36.652172
Saurav   4200     130      9       32.307693
ThePhenomenalRNB          8888     105      18      84.647621
```

3 Electricity

Problem 1.3 *Calculate electric charge for the following rates*
For first 100 units *60P per unit*
For next 200 units *80P per unit*
Beyond 300 units *90P per unit*
Minimum charge is Rs. 50.00. If total amount is more than 300.00, additional 15% charge is added.
Read names of users and units consumed and print the charge with names

```
1
22
333
4444
55555
...
```

Code.

```
#include<stdio.h>
using namespace std;
```

```
typedef struct charge {
        char name[50];
        int units;
        float cost;
}charge;

float findCost(int n)
{
        float c=0;
        if(n>=100) {
                c+=(100*0.6);
                n-=100;
        } else {
                c+=(n*0.6);
                return c;
        }
        if(n>=200) {
                c+=(200*0.8);
                n-=200;
        } else {
                c+=(n*0.8);
                return c;
        }
        if(n>0) {
                c+=(n*0.9);
                return c;
        }
}

int main()
{
        int n,i;
        scanf("%d",&n);
        charge chs[n];
        for(i=0;i<n;i++) {
                printf("Enter name:");
                scanf("%s",chs[i].name);
                printf("Enter no of units for %s\n",chs[i].name);
                scanf("%d",&chs[i].units);
                chs[i].cost=500.0;
                chs[i].cost+=findCost(chs[i].units);
                if(chs[i].cost>300) {
                        chs[i].cost+=(0.15*chs[i].cost);
                }
        }
        for(i=0;i<n;i++) {
                printf("%s\t%d\t%f\n",chs[i].name,
                chs[i].units,chs[i].cost);
        }
```

```
        return 0;
}
```

Output.

```
3
Enter name:Rudra
Enter no of units for Rudra
250
Enter name:Tokon
Enter no of units for Tokon
10
Enter name:Rohit
Enter no of units for Rohit
300
Rudra   250     782.000000
Tokon   10      581.900024
Rohit   300     828.000000
```

4 Election

Problem 1.4 *An election is contested by five candidates, numbered 1-5. Voting is done on ballot paper. Write a program to read the ballots and count the votes for each candidates. Any vote outside the range 1-5 is "split vote". Count the split votes as well*

Code.

```
#include<stdio.h>
#include<string.h>
#include<algorithm>
#include<vector>
#include<queue>
#include<map>
#include<math.h>

#define ll long long int

int max(int a, int b)
{
        if(a>=b)
                return a;
        return b;
}

using namespace std;

int main()
{
        int n,i,count=0;
        int hash[7];
        memset(hash,0,sizeof(hash));
```

```
        scanf("%d",&n);
        while(n--) {
                count++;
                printf("Whom did %d vote for ? \n",count);
                int vote;
                scanf("%d",&vote);
                if(vote>=1 && vote<=5) {
                        hash[vote]++;
                } else {
                        hash[6]++;
                }
        }
        for(i=1;i<=5;i++) {
                printf("No of people voted for %d = %d\n",i,hash[i]);
        }
        printf("No of invalid votes = %d\n",hash[6]);
        return 0;
}
```

Output.

```
8
Whom did 1 vote for ?
1
Whom did 2 vote for ?
1
Whom did 3 vote for ?
2
Whom did 4 vote for ?
1
Whom did 5 vote for ?
5
Whom did 6 vote for ?
9
Whom did 7 vote for ?
2
Whom did 8 vote for ?
1
No of people voted for 1 = 4
No of people voted for 2 = 2
No of people voted for 3 = 0
No of people voted for 4 = 0
No of people voted for 5 = 1
No of invalid votes = 1
```

5 Factorial

Problem 2.1 *Calculate factorial of a number in C++ using functions*

Code.

```
#include<iostream>
```

```
using namespace std;

int fact(int n)
{
        if(n==1)
                return 1;
        return n*fact(n-1);
}

int main()
{
        int n;
        cout<<"Enter number"<<endl;
        cin>>n;
        cout<<"Factorial of "<<n<<": "<<fact(n)<<endl;
        return 0;
}
```

Output.

```
Enter number
5
Factorial of 5: 120
```

6 Series sum

Problem 2.2 *Calculate the sum of the series 1+22+32+42+... nth term in C++ using functions*

Code.

```
#include<iostream>

using namespace std;

int ser(int n)
{
        int m=1,sum=1,no=12;
        while(m<=n) {
                m++;
                sum+=no;
                no+=10;
        }
        return sum;
}

int main()
{
        cout<<"n: ";
        int n;
        cin>>n;
```

```
        cout<<"Sum of series: "<<ser(n)<<endl;
        return 0;
}
```

Output.

```
n: 5
Sum of series: 161
```

7 Array search

Problem 2.3 *Find the smallest and the largest no in an array and search for an element in the array in C++ using functions*

Code.

```
#include<stdio.h>
#include<iostream>
#include<algorithm>

using namespace std;

void search(int a[], int n, int srch)
{
        int lo=0,hi=n-1,mid;
        while(lo<=hi) {
                mid=(lo+hi)/2;
                if(a[mid]==srch) {
                        cout<<"Found at position: "<<(mid+1)<<endl;
                        return;
                } else if(a[mid]>srch) {
                        hi=mid-1;
                } else if(a[mid]<srch) {
                        lo=mid+1;
                }
        }
        cout<<"Not Found\n";
}

int main()
{
        int n,i;
        cout<<"n: ";
        cin>>n;
        int a[n];
        for(i=0;i<n;i++) {
                cout<<"a["<<i<<"]: ";
                cin>>a[i];
        }
        sort(a,a+n);
        cout<<"Mininum No: "<<a[0]<<endl;
        cout<<"Maximum No: "<<a[n-1]<<endl;
```

```
        int srch;
        cout<<"No to search: ";
        cin>>srch;
        search(a, n, srch);
}
```

Output.

```
n: 5
a[0]: 5
a[1]: 4
a[2]: 3
a[3]: 2
a[4]: 1
Mininum No: 1
Maximum No: 5
No to search: 4
Found at position: 4
```

8 Matrix Multiplication

Problem 2.4 *Multiply two matrix in C++*

Code.

```
#include<stdlib.h>
#include <stdio.h>
#include<time.h>

using namespace std;

int main()
{

        int m,n,c,p,q,d,k,sum=0;
        int first[10][10]; // maximum upto 10 X 10 Matrix
        int second[10][10];
        int multiply[10][10]; // Final result will be stored here
        printf("Enter number of rows and Columns of first matrix\n");
        scanf("%d %d",&m,&n);
        int i,j;
        srand(time(NULL)); // Starting the seed
        for(i=0;i<m;i++)   // replace with random number
        {
                for(j=0;j<n;j++)
                {

                        first[i][j]=rand();
                        first[i][j]=first[i][j]%11;
                }
        }
        printf("Enter number of rows and Columns of second matrix\n");
```

```
        scanf("%d %d",&p,&q);
        for(i=0;i<p;i++)     // replace with random number
        {
                for(j=0;j<q;j++)
                {
                        second[i][j]=rand();
                        second[i][j]=second[i][j]%11;
                }
        }
        if(n!=p)    // the error message if not compatible
        {
                printf("\nERROR:CANNOT MULTIPLY\n");
        }
        else
        {

                printf("\n---------------\n");
                printf("\nFIRST MATRIX IS : \n");
                for(i=0;i<m;i++)  // printing first matrix
                {
                        for(j=0;j<n;j++)
                        {
                                printf("%d\t",first[i][j]);
                        }
                        printf("\n");
                }
                printf("\nSECOND MATRIX IS : \n");
                for(i=0;i<p;i++) // printing second matrix
                {
                        for(j=0;j<q;j++)
                        {
                                printf("%d\t",second[i][j]);
                        }
                        printf("\n");
                }
                for(c=0;c<m;c++) // m,q,p
                {
                        for(d=0;d<q;d++)
                        {
                                for(k=0;k<p;k++)
                                {
                                        sum=sum+ first[c][k]*second[k][d];
                                }
                                multiply[c][d]=sum;
                                sum=0;
                        }
                }
                printf("\nMULTIPLIED MATRIX IS : \n");
                for(c=0;c<m;c++)//m,q
                {
```

```
                        for(d=0;d<q;d++)
                        {
                                printf("%d\t",multiply[c][d]);
                        }
                        printf("\n");
                }
        }
        return 0;
}
```

Output.

```
Enter number of rows and Columns of first matrix
2
3
Enter number of rows and Columns of second matrix
3
2
---------------

FIRST MATRIX IS :
10      0       8
8       10      3

SECOND MATRIX IS :
6       5
3       8
8       1

MULTIPLIED MATRIX IS :
124     58
102     123
```

9 Students with unique roll numbers

Problem 3.1 *Create objects of Student class such that all students have different roll numbers*
Code.

```
#include<stdio.h>
using namespace std;
class student {
        int roll;
public:
        static int z;
        student() {}
        void init()
        {
                roll=z++;
        }
        void display()
        {
```

```
                printf("Roll %d\n",roll);
        }
};
int student::z;

int main()
{
        student st[10];
        int i;
        for(i=0;i<10;i++) {
                int x=student::z;
                st[i].init();
        }
        printf("The rolls are\n");
        for(i=0;i<10;i++) {
                st[i].display();
        }
        return 0;
}
```

Output.

```
The rolls are
Roll 0
Roll 1
Roll 2
Roll 3
Roll 4
Roll 5
Roll 6
Roll 7
Roll 8
Roll 9
```

10 Complex Number Addition

Problem 3.2 *Add two complex numbers*

Code.

```
#include<stdio.h>
using namespace std;
class Complex
{
        int real;
        int img;
        public:
                Complex() {
                        real=0;
                        img=0;
                }
                Complex(int real, int img)
```

```
                {
                        this->real=real;
                        this->img=img;
                }
                static Complex add(Complex a, Complex b)
                {
                        Complex c;
                        c.real=a.real+b.real;
                        c.img=a.img+b.img;
                        return c;
                }
                void show()
                {
                        printf("%d+i%d\n",real,img);
                }
        private:
};
int main()
{
        Complex a(10,20);
        Complex b(5,-5);
        printf("No.s to add\n");
        a.show();
        b.show();
        Complex c;
        c=Complex::add(a,b);
        printf("Result: ");
        c.show();
}
```

Output.

```
No.s to add
10+i20
5+i-5
Result: 15+i15
```

11 Friend Function

Problem 3.3 *Swap values of 2 objects of two different class using* ***friend*** *function*
Code.

```
#include<stdio.h>
#define SWAP(A,B) (A)=((A)+(B))-((B)=(A))
using namespace std;
class A
{
        public:
                int val;
                A(int _val)
                {
```

```
                        val=_val;
                }
                void show()
                {
                        printf("Value:_%d\n",val);
                }
        private:
                friend void swap(A, A);
};

class B
{
        public:
                B() {}
                B(A *a, A *b)
                {}
        void swap(A *a, A *b)
        {
                SWAP(a->val, b->val);
        }
};

int main()
{
        A a(10);
        A b(20);
        a.show();
        b.show();

        B sw(&a,&b);
        sw.swap(&a, &b);

        printf("After_swapping...\n");

        a.show();
        b.show();
}
```

Output.

```
Value: 10
Value: 20
After swapping...
Value: 20
Value: 10
```

12 Distance

Problem 3.4 *Write a program to take distances as input, one in inches, other in metres and to convert the distances in inches and add.*

Code.

```
#include<stdio.h>

class DM {
        int m;
        int cm;
        public:
        DM() {}
        DM(int m, int cm)
        {
                this->m=m;
                this->cm=cm;
        }
        friend float convert(DM);
};

class DB {
        float inch;
        int ft;
        public:
        DB() {}
        DB(int inch, int ft)
        {
                this->inch=inch;
                this->ft=ft;
        }
        friend float convert(DM a)
        {
                return a.cm*2.25f;
        }
        void add(DM a)
        {
                inch+=convert(a);
                printf("Inc=%f\n",inch);
        }
};

int main()
{
        DM a(1,100);
        DB b(2,60);
        b.add(a);
        return 0;
}
```

Output.

```
Inc=227.000000
```

13 User defined String class

Problem 3.5 *Define a class **String**, that could work as a user defined string type. Include constructors to initialise an uninitialised string,*
String s1; // string with length 0
and also initialise a string with constant value
String s2("Well done");
*Write a program to create and concatenate two objects of this **String** class.*

Code.

```
#include<stdio.h>
#include<iostream>
#include<string>
#include<cstring>

using namespace std;

class String {
        string s;
        public:
        String() {
                s="";
        }
        String(string str)
        {
                s=str;
        }
        void concat(string s2)
        {
                s=s+s2;
        }
        void disp()
        {
                cout<<s<<endl;
        }
};
int main()
{
        String s("Rudra");
        s.disp();
        s.concat("NilBasu");
        s.disp();
        return 0;
}
```

Output.

```
Rudra
RudraNilBasu
```