**Phase-4 Array Programs**

**Practical-1**

**Aim: A Train going to Busan have 2 containers which contains Zombies. Container A has 6 zombies, and Container B has 4 zombies. Passengers have to reach in engine container by passing through them. Help them by transferring zombies from both that containers to a new Container C. Build a C++ program for it.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Zombies

{

private:

int a[3]={1,3,5};

int b[7]={2,4,6,7,8,9,10};

int c[10],i;

public:

void Train()

{

cout<<" Container A has 3 zombies" <<

" and Container B has 7 zombies."<<endl;

cout<<"\* A = ";

for(i=0;i<3;i++)

{

cout<<", "<<a[i];

}

cout<<endl<<"\* B = ";

for(i=0;i<7;i++)

{

cout<<", "<<b[i];

}

for(i=0;i<3;i++)

{

c[i]=a[i];

}

for(i=0;i<7;i++)

{

c[3+i]=b[i];

}

cout<<endl<<endl<<"=> Transferring zombies from" <<

" both containers to C Container."<<endl;

cout<<"\* C = ";

for(i=0;i<10;i++)

{

cout<<", "<<c[i];

}

cout<<endl;

}

};

int main()

{

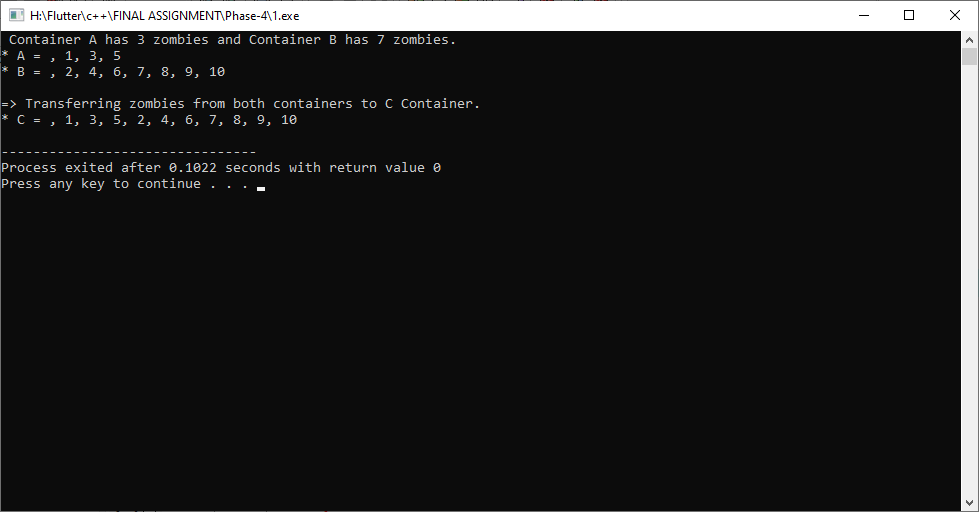
Zombies z1;

z1.Train();

return 0;

}

**Output:**

****

**Practical-2**

**Aim: Hitler ordered a 10 soldiers pared to align in a row. He wants to know that which soldier have the highest killing score. Help him by design a C++ Program.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Hitler

{

private:

int a[5][2]={{1,2},

{2,3},

{3,4},

{4,5},

{5,6}};

int i,j;

public:

void Kill()

{

for(i=0;i<5;i++)

{

for(j=0;j<2;j++)

{

if(a[i][j]>a[0][0])

{

a[0][0]=a[i][j];

}

}

}

cout<<"=> Highest killing score is :- "<<a[0][0]<<endl;

}

};

int main()

{

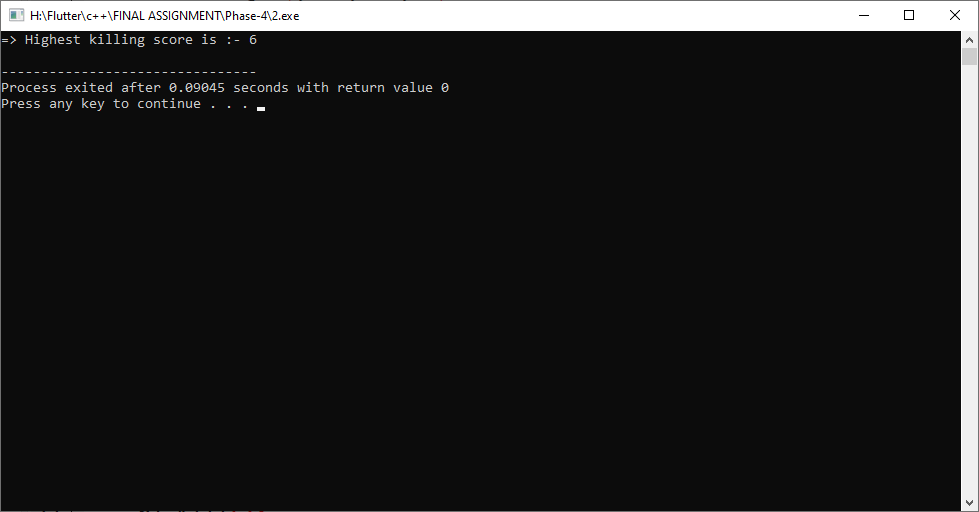
Hitler h1;

h1.Kill();

return 0;

}

**Output:**

****

**Practical-3**

**Aim: Design a C++ system which automatically identifies if a given word contains any letter or symbol between both SHIFT keys from our regular PC Keyboard.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Keyboard

{

private:

char a[100];

int i;

public:

Keyboard()

{

cout<<"\* Enter any string :- "; cin>>this->a;

}

void Shift()

{

cout<<endl<<"=> All letter or symbol between both SHIFT keys is :- ";

for(i=0;a[i]!='\0';i++)

{

if(a[i]=='z'||a[i]=='x'||a[i]=='c'||a[i]=='v'||a[i]=='b'||a[i]=='n'||a[i]=='m'||

a[i]=='Z'||a[i]=='X'||a[i]=='C'||a[i]=='V'||a[i]=='B'||a[i]=='N'||a[i]=='M'||

a[i]==','||a[i]=='.'||a[i]=='/'||a[i]=='<'||a[i]=='>'||a[i]=='?')

{

cout<<a[i];

}

}

}

};

int main()

{

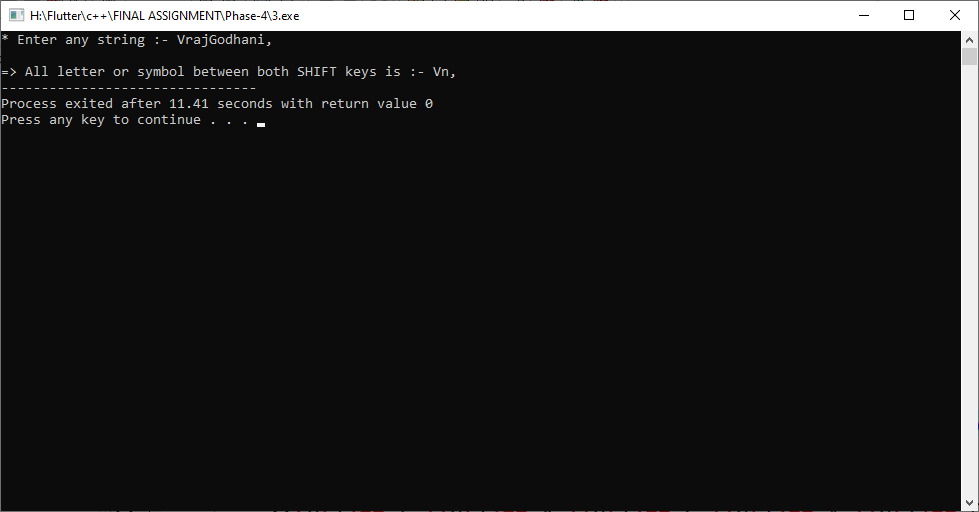
Keyboard k1;

k1.Shift();

return 0;

}

**Output:**

****

**Practical-4**

**Aim: Ronak gives an examination in which he gains internal viva marks in all subjects like this: 23, 19, 22, 28 & 23 (all marks are out of 30). And gains final written examination marks like this: 65, 58, 49, 52 & 64 (all marks are out of 70). Evaluate final total examination marks by adding both exam marks and reveal marks out of 100 in each subject by using C++. Also, final total average value for that.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Ronak

{

private:

int Ronak1[5] = {23, 19, 22, 28, 23};

int Ronak2[5] = {65, 58, 49, 52, 64};

int sum=0, i;

public:

void Mark()

{

cout<<endl<<"\* Ronak viva marks :- 23, 19, 22, 28, 23"<<endl;

cout<<"\* Ronak examination marks :- 65, 58, 49, 52, 64 "<<endl;

cout<<endl<<"=> Final Examination Mark of :- ";

for(i=0;i<5;i++)

{

cout<<" , "<<Ronak1[i]+Ronak2[i];

sum += (Ronak1[i]+Ronak2[i]);

}

cout<<endl;

cout<<endl<<"=> Average of all marks :- "<<sum/5<<endl;

}

};

int main()

{

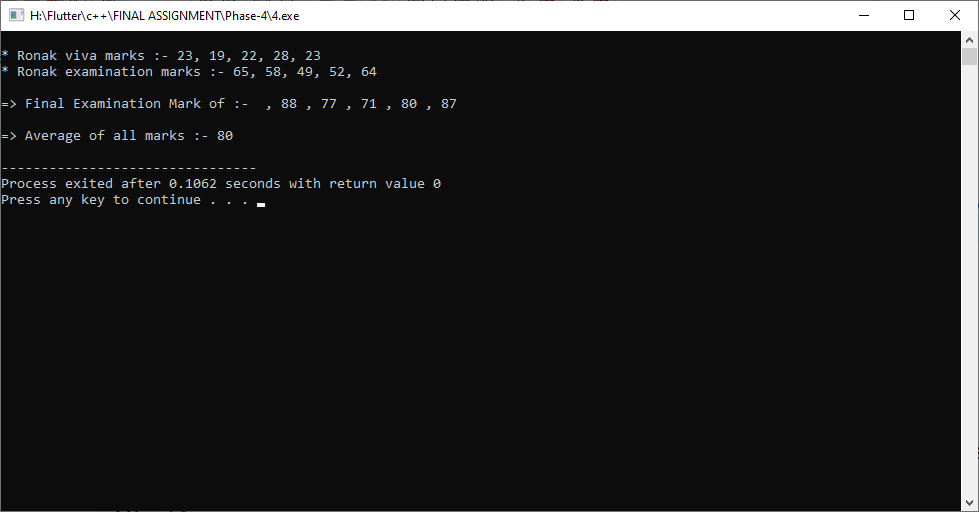
Ronak r1;

r1.Mark();

return 0;

}

**Output:**

****

**Practical-5**

**Aim: Devansh gives a list of random numbers to his colleague Rohan to distinguish all odd and even numbers, and store them in different lists. Help Rohan by building such a solution with help of C++.**

**Program:**

#include<iostream>

#include<stdio.h>

using namespace std;

class Array

{

private:

int a[100], b[100], c[100];

int n,j=0,t=0,i,k;

public:

Array()

{

cout<<endl<<" Enter How many Number: ";

cin>>n;

cout<<endl;

for(i=0,k=1;i<n;i++,k++)

{

cout<<" Enter Number ["<<k<<"] : ";

cin>>a[i];

}

}

void solution()

{

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

{

if(a[i]%2==0)

{

b[j]=a[i];

j++;

}

else if(a[i]%2==1)

{

c[t]=a[i];

t++;

}

}

cout<<endl<<"-----------------------------------------------------------"<<endl;

cout<<" All Even Number is: ";

for(i=0;i<j;i++)

{

cout<<b[i]<<" , ";

}

cout<<endl;

cout<<"-----------------------------------------------------------"<<endl;

cout<<endl<<"-----------------------------------------------------------"<<endl;

cout<<" All Odd Number is: ";

for(i=0;i<t;i++)

{

cout<<c[i]<<" , ";

}

cout<<endl;

cout<<"-----------------------------------------------------------"<<endl;

}

};

int main()

{

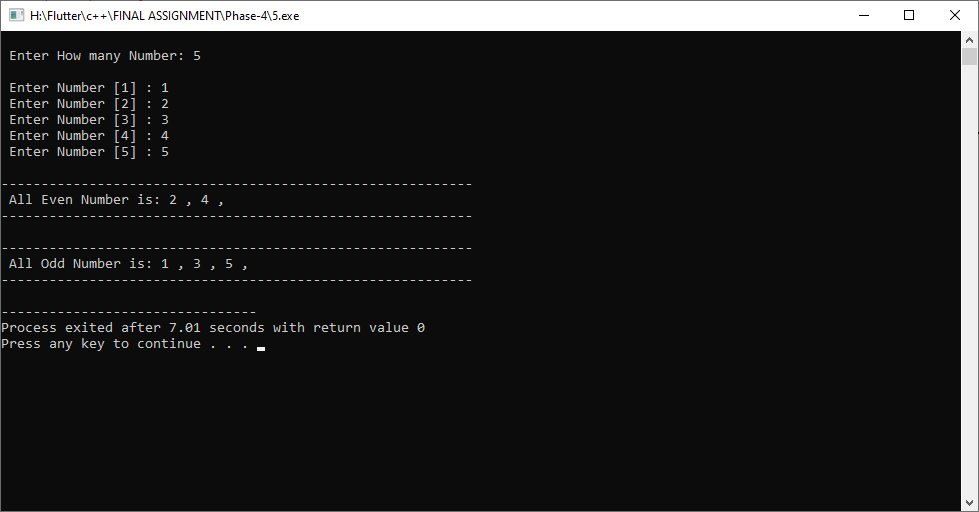
Array a1;

a1.solution();

return 0;

}

**Output:**

****

**Practical-6**

**Aim: A Teacher gives a list to students in which a list contains many years in numeric format i.e 1994, 1947, 2002, 1996, etc. All students supposed to find all duplicate years which occurs more than once and store them into another list. Design such type of system with help of C++.**

**Program:**

#include<iostream>

using namespace std;

class Teacher

{

private:

int y[20];

int i,j,n=1;

public:

Teacher()

{

cout<<endl<<" Enter 20 Year List: "<<endl;

cout<<"======================================="<<endl;

for(i=0;i<20;i++)

{

cout<<"Year "<<n<<" : ";

cin>>y[i];

n++;

}

}

void solution()

{

cout<<endl<<" Dublicate Year List: "<<endl;

cout<<"======================================="<<endl;

for(i=0;i<20;i++)

{

if(y[i]!=-1)

{

for(j=i+1;j<20;j++)

{

if(y[i]==y[j])

{

y[j]=-1;

}

}

}

}

for(i=0;i<20;i++)

{

if(y[i]!=-1)

{

cout<<y[i]<<" , ";

}

}

}

};

int main()

{

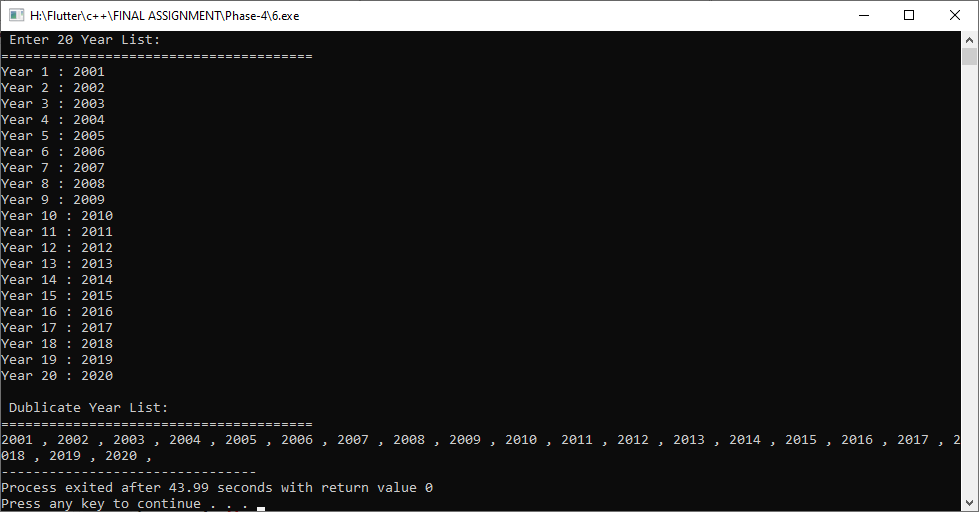
Teacher t1;

t1.solution();

return 0;

}

**Output:**

****

**Practical-7**

**Aim: A Frontman can randomly assign two 3x3 matrices to all participants in Squid Games. All participants have to add that matrices and store final answer as a separate matrix to win this type of round in the game. Build a C++ system to help them all.**

**Program:**

#include<iostream>

using namespace std;

class Squid\_Games

{

private:

int a[3][3], b[3][3], c[3][3];

int i,j;

public:

void setdata()

{

cout<<endl<<" Enter First Matrix Value:-"<<endl;

cout<<"============================================"<<endl;

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

cout<<" a["<<i<<"]["<<j<<"]: ";

cin>>a[i][j];

}

}

cout<<endl<<" Enter Second Matrix Value:-"<<endl;

cout<<"============================================"<<endl;

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

cout<<" b["<<i<<"]["<<j<<"]: ";

cin>>b[i][j];

}

}

}

void solution()

{

cout<<endl<<" Addition Matrix is:-"<<endl;

cout<<"============================================"<<endl;

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

c[i][j]=a[i][j]+b[i][j];

cout<<c[i][j]<<" ";

}

cout<<endl;

}

}

};

int main()

{

Squid\_Games s1;

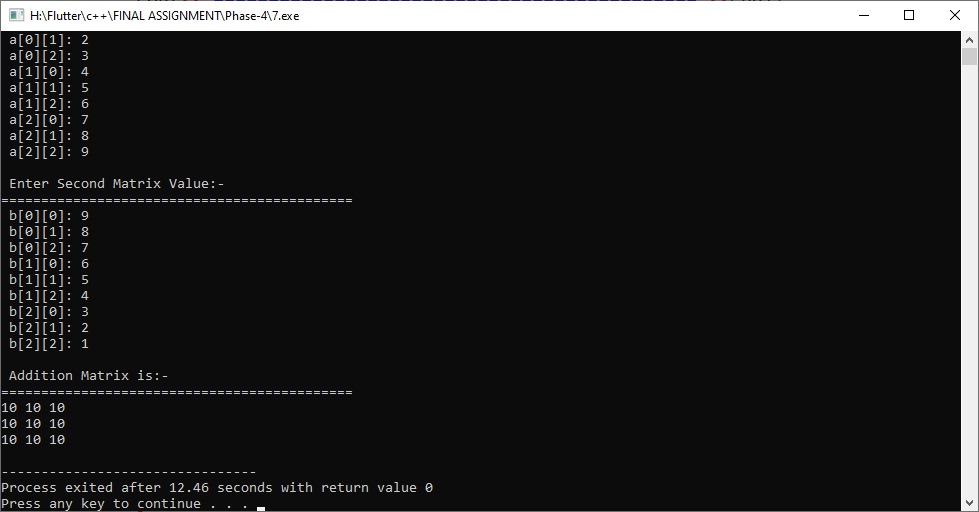
s1.setdata();

s1.solution();

return 0;

}

**Output:**

****

**Practical-8**

**Aim: Design a system in C++ which automatically transpose any given Matrix of RxC dimension. Where R is number of Rows and C is number of Columns. Help three musketeers for passing an interview round by solving this last question.**

**Program:**

#include<iostream>

using namespace std;

class musketeers

{

private:

int a[100][100];

int c,r,i,j;

public:

void setdata()

{

cout<<endl<<" Enter How Many Rows: ";

cin>>r;

cout<<" Enter How Many Columns: ";

cin>>c;

cout<<endl;

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

cout<<" a["<<i<<"]["<<j<<"]: ";

cin>>a[i][j];

}

}

}

void solution()

{

cout<<endl<<" Matrix is:-"<<endl;

cout<<"============================================"<<endl;

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

cout<<a[i][j];

}

cout<<endl;

}

cout<<endl<<" Transpose Matrix is:-"<<endl;

cout<<"============================================"<<endl;

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

cout<<a[j][i];

}

cout<<endl;

}

}

};

int main()

{

musketeers m1;

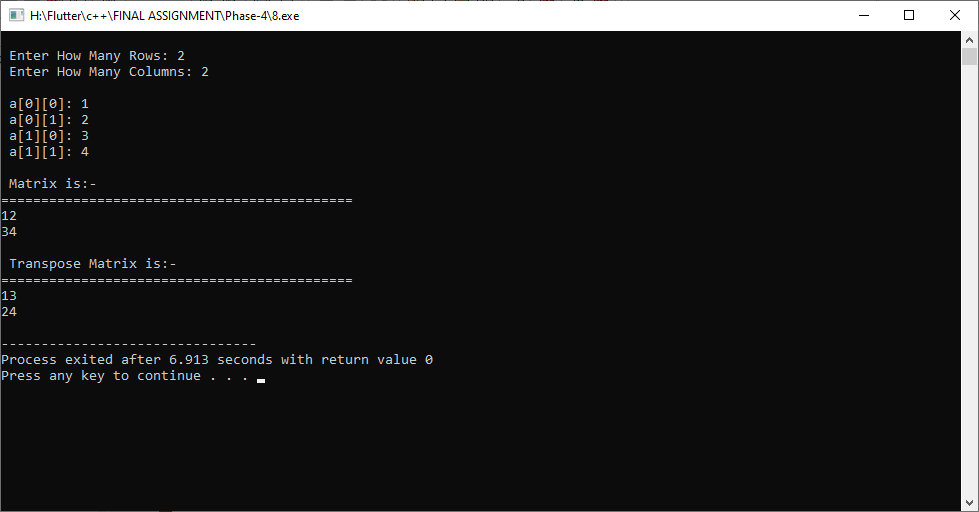
m1.setdata();

m1.solution();

return 0;

}

**Output:**

****

**Practical-9**

**Aim: Columbian army arranged all 9 forbidden refugees in a 3x3 matrix formation. An army cadets have to find that which one of the refugees stats the highest weight and which one weighs the lowest weight. Help army cadets by preparing C++ solution for their undercover mission.**

**Program:**

#include<iostream>

using namespace std;

class Columbian

{

private:

int a[3][3];

int i,j;

public:

void setdata()

{

cout<<endl<<" Enter refugees stats weight:-"<<endl;

cout<<"============================================"<<endl;

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

cout<<" a["<<i<<"]["<<j<<"]: ";

cin>>a[i][j];

}

}

}

void solution()

{

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

if(a[i][j]>a[0][0])

{

a[0][0]=a[i][j];

}

}

}

cout<<endl<<"--------------------------------------------------------"<<endl;

cout<<" One of the refugees stats the highest weight is: "<<a[0][0]<<endl;

cout<<"-------------------------------------------------------"<<endl;

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

if(a[i][j]<a[0][0])

{

a[0][0]=a[i][j];

}

}

}

cout<<endl<<"--------------------------------------------------------"<<endl;

cout<<" One of the refugees stats the Lowest weight is: "<<a[0][0]<<endl;

cout<<"-------------------------------------------------------"<<endl;

}

};

int main()

{

Columbian c1;

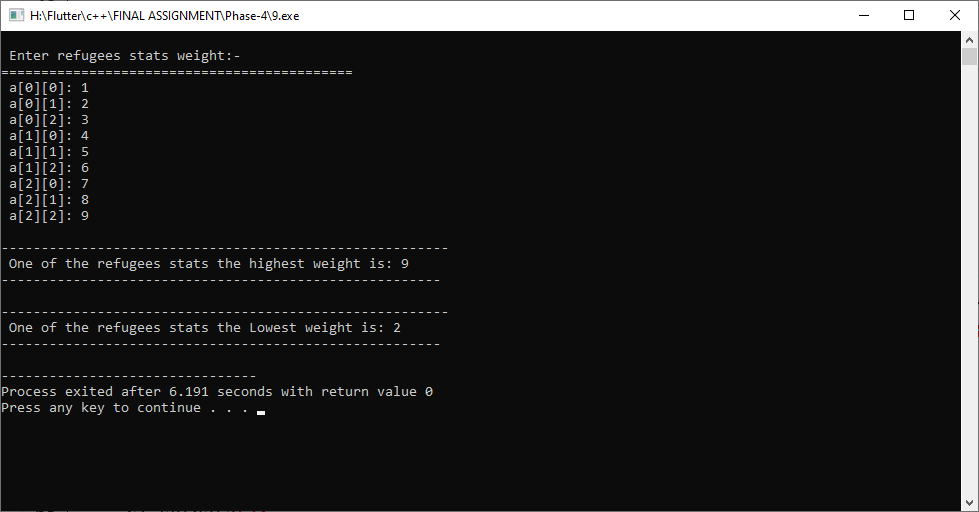
c1.setdata();

c1.solution();

return 0;

}

**Output:**

****

**Practical-10**

**Aim: Help Martin to solve a special kind of puzzle by designing a C++ system. Total 25 random numbers arranged in a form of Square Matrix. To solve that puzzle, he has to find addition of all diagonally aligned numbers on puzzle cardboard.**

**Program:**

#include<iostream>

using namespace std;

class Martin

{

private:

int a[5][5];

int i,j,sum=0;

public:

Martin()

{

cout<<endl<<" Enter 25 Random Number:-"<<endl;

cout<<"============================================"<<endl;

for(i=0;i<5;i++)

{

for(j=0;j<5;j++)

{

cout<<" a["<<i<<"]["<<j<<"]: ";

cin>>a[i][j];

}

}

}

void solution()

{

cout<<endl<<" Matrix is:-"<<endl;

cout<<"============================================"<<endl;

for(i=0;i<5;i++)

{

for(j=0;j<5;j++)

{

cout<<a[i][j]<<" ";

}

cout<<endl;

}

cout<<"============================================"<<endl;

for(i=0;i<5;i++)

{

for(j=0;j<5;j++)

{

if(i==j)

{

sum+=a[i][j];

}

}

}

cout<<endl<<"--------------------------------------------------------"<<endl;

cout<<" Addition of Digonal Elements is: "<<sum<<endl;

cout<<"-------------------------------------------------------"<<endl;

}

};

int main()

{

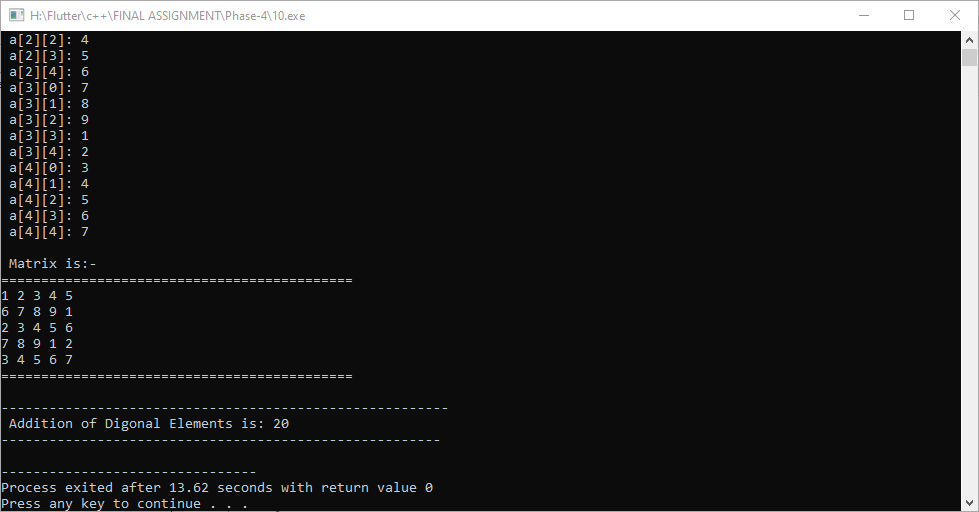
Martin m1;

m1.solution();

return 0;

}

**Output:**

****

**Practical-11**

**Aim: A one-sided open Tennis ball jar has capacity of storing total 5 different balls. Each ball has unique number attached as a label itself. Arrange all that balls in a jar in such a way that their order is stats as a reverse by referring attached numbers as a label. Create a C++ system for doing such type of task.**

**Program:**

#include<iostream>

using namespace std;

class Jar

{

private:

int j[5]={111, 222, 333, 444, 555};

int i;

public:

void solution()

{

cout<<endl<<" Every Ball with unique Number: "<<endl;

cout<<"========================================"<<endl;

for(i=0;i<5;i++)

{

cout<<" Ball Number is: "<<i<<" And Ball is: "<<j[i]<<endl;;

}

cout<<endl<<" Ball order is stats as a reverse: "<<endl;

cout<<"========================================"<<endl;

for(i=4;i>=0;i--)

{

cout<<" Ball Number is: "<<i<<" And Ball is: "<<j[i]<<endl;;

}

}

};

int main()

{

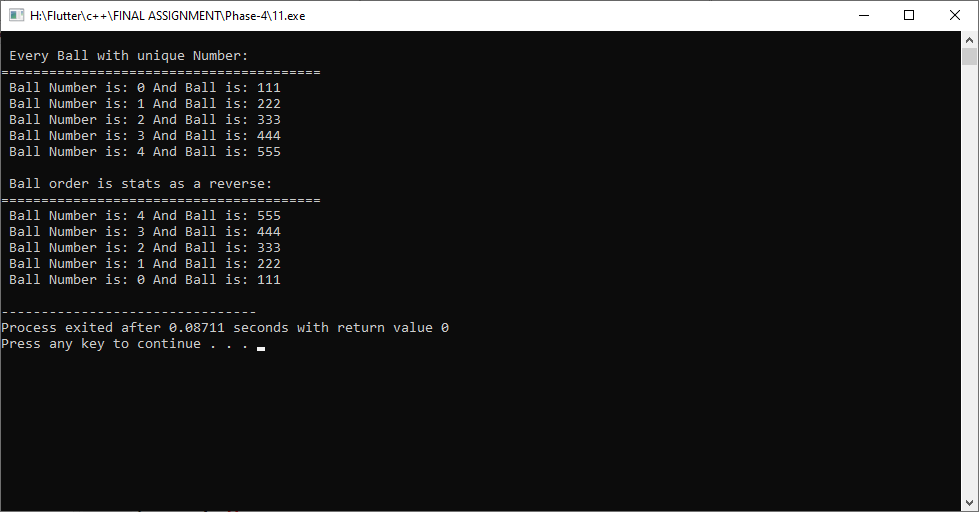
Jar j1;

j1.solution();

return 0;

}

**Output:**

****

**Practical-12**

**Aim:A College wants to celebrate all degree holder students to throwing their hats in a predefined way: First all 25 students have to arranged in a Square Matrix. First, an upper half of triangle matrix will throwing hats and then a lower half of triangle matrix will. Help them to achieve this unique idea by using C++. 13. A Math teacher wants to teach how**

**Program:**

#include<iostream>

using namespace std;

class College

{

private:

int a[5][5]={{1,1,1,1,1},

{2,2,2,2,2},

{3,3,3,3,3},

{4,4,4,4,4},

{5,5,5,5,5}

};

int i,j,k;

public:

void solution()

{

cout<<endl<<" Student Matrix.."<<endl;

cout<<"============================================"<<endl;

for(i=0;i<5;i++)

{

for(j=0;j<5;j++)

{

cout<<a[i][j]<<" ";

}

cout<<endl;

}

cout<<endl<<" First, Showing position student throwing hats.."<<endl;

cout<<"======================================================"<<endl;

for(i=0;i<5;i++)

{

for(j=0,k=i;j<5;j++,k++)

{

if(k<=4)

{

cout<<a[i][k]<<" ";

}

}

cout<<endl;

}

cout<<endl<<" Secondly, Showing position student throwing hats.."<<endl;

cout<<"======================================================"<<endl;

for(i=0;i<5;i++)

{

for(j=0,k=0;j<5;j++,k++)

{

if(k<=i)

{

cout<<a[i][k]<<" ";

}

}

cout<<endl;

}

}

};

int main()

{

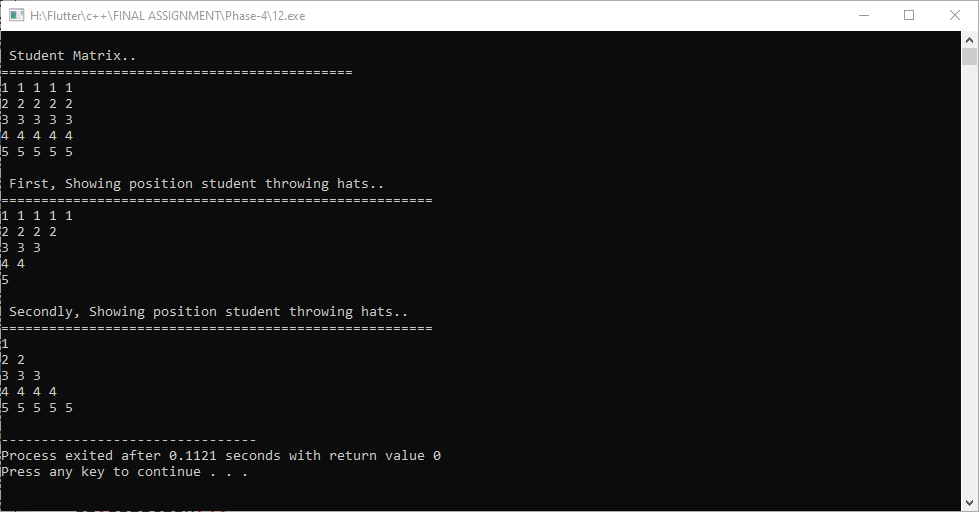
College c1;

c1.solution();

return 0;

}

**Output:**



**Practical-13**

**Aim:A Maths teacher wants to teach how to perform a dot product of two matrices. Design a better approach in C++ to help this maths teacher.**

**Program:**

#include<iostream>

using namespace std;

class Math\_teacher

{

private:

int a[100][100], b[100][100],d[100][100];

int r,c,i,j,r1,c1,k;

public:

Math\_teacher()

{

cout<<endl<<" Enter Detail of First Matrix: "<<endl;

cout<<"============================================"<<endl;

cout<<endl<<" Enter How Many Rows: ";

cin>>r;

cout<<" Enter How Many Columns: ";

cin>>c;

cout<<endl;

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

cout<<" a["<<i<<"]["<<j<<"]: ";

cin>>a[i][j];

}

}

cout<<endl<<" Enter Detail of Second Matrix: "<<endl;

cout<<"============================================"<<endl;

cout<<endl<<" Enter How Many Rows: ";

cin>>r1;

cout<<" Enter How Many Columns: ";

cin>>c1;

cout<<endl;

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

{

cout<<" b["<<i<<"]["<<j<<"]: ";

cin>>b[i][j];

}

}

}

void solution()

{

if(r==r1&&c==c1)

{

cout<<endl<<" Dot product of above Two Matrix is:"<<endl;

cout<<"======================================================"<<endl;

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

d[i][j]=0;

for(k=0;k<r;k++)

{

d[i][j]+=a[i][k]\*b[k][j];

}

cout<<d[i][j]<<" ";

}

cout<<endl;

}

}

else

{

cout<<endl<<" Dot Production is n't Posible.."<<endl;

}

}

};

int main()

{

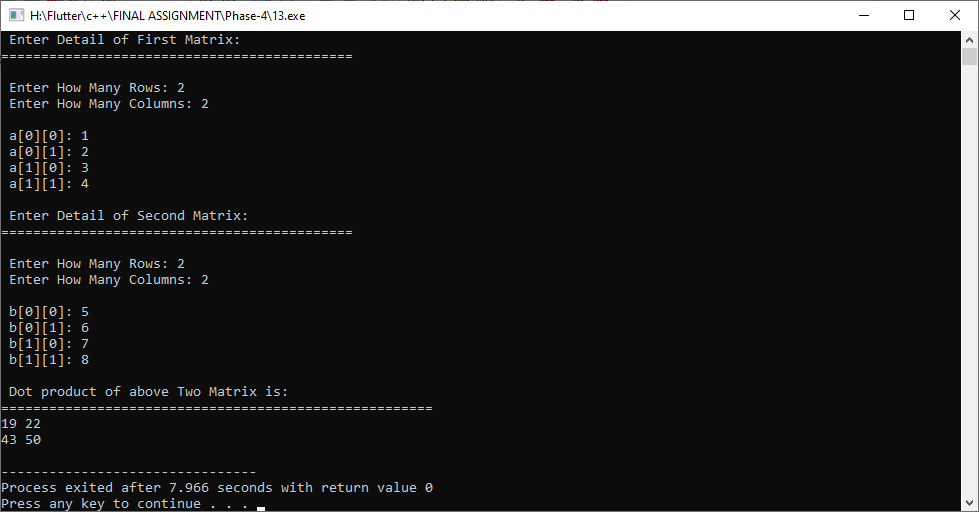
Math\_teacher m1;

m1.solution();

return 0;

}

**Output:**

****