

Assignment :3

IT-603

Name:- Rajpurohit Jeevan Bhawarlal

Student ID-202312090

1).

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main() {
```

```
    int m = 2, n = 3, l = 2;
```

```
    int a[m][n] = {{1, 2, 3}, {4, 5, 6}};
```

```
    int b[n][l] = {{1, 2}, {3, 4}, {5, 6}};
```

```
    int c[m][l] ;
```

```
    for (int i = 0; i < m; i++) {
```

```
        for (int j = 0; j < l; j++) {
```

```
            c[i][j]=0;
```

```
            for (int k = 0; k < n; k++) {
```

```
                c[i][j] += a[i][k] * b[k][j];
```

```
            }
```

```
        }
```

```
    }
```

```

for (int i = 0; i < m; i++) {
    for (int j = 0; j < l; j++) {
        cout << c[i][j] << " ";
    }
    cout << endl;
}

return 0;
}

```

```

main.cpp
6  int a[m][n] = {{1, 2, 3}, {4, 5, 6}};
7  int b[n][l] = {{1, 2}, {3, 4}, {5, 6}};
8  int c[m][l];
9
10 for (int i = 0; i < m; i++) {
11     for (int j = 0; j < l; j++) {
12         c[i][j] = 0;
13         for (int k = 0; k < n; k++) {
14             c[i][j] += a[i][k] * b[k][j];
15         }
16     }
17 }
18
19 for (int i = 0; i < m; i++) {
20     for (int j = 0; j < l; j++) {
21         cout << c[i][j] << " ";
22     }
23     cout << endl;
24 }
25
26 return 0;
27 }
28
input
22 28
49 64
...Program finished with exit code 0
Press ENTER to exit console.

```

2).

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```

int main(){

    float matrix[3][3]={1,2,3},{1,0,3},{1,2,0}},mat[3][3];

    float d_matrix=0;

    for(int a=0,b=0 ; a<3 ; a++,b++){

        d_matrix+= matrix[0][a]*(matrix[1][(b+1)%3]*matrix[2][(b+2)%3]-
matrix[1][(b+2)%3]*matrix[2][(b+1)%3]);

    }

    if(d_matrix==0){

        cout<<"Inverse of given matrix cannot be find as determinant is 0"<<endl;

    }

    else{

        float inverse_matrix[3][3];

        for(int a=0 ; a<3 ; a++){

            for(int b=0 ; b < 3 ; b++){

                inverse_matrix[b][a]=
((matrix[(a+1)%3][(b+1)%3]*matrix[(a+2)%3][(b+2)%3])-
(matrix[(a+1)%3][(b+2)%3]*matrix[(a+2)%3][(b+1)%3]))/d_matrix;

            }

        }

        for(int a=0 ; a<3 ; a++){

            for(int b=0 ; b < 3 ; b++){

```

```

        cout<<inverse_matrix[a][b]<<" ";

    }

    cout<<endl;

}

}

}

```

The screenshot shows the OnlineGDB interface with a C++ program. The code calculates the determinant of a 3x3 matrix and then finds its inverse using cofactors. The input matrix is:
 
$$\begin{bmatrix} 1 & 1 & 1 \\ 0.5 & -0.5 & 0 \\ 0.333333 & 0 & -0.333333 \end{bmatrix}$$
 The output shows the inverse matrix:
 
$$\begin{bmatrix} -1 & 1 & 1 \\ 0.5 & -0.5 & 0 \\ 0.333333 & 0 & -0.333333 \end{bmatrix}$$
 The program finishes with exit code 0.

3).

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```

int main(){

    float matrix[3][3]={1,1,1},{0,0,1},{0,1,1}};

    float d_matrix=0;

```

```

for(int a=0,b=0 ; a<3 ; a++,b++){

    d_matrix+= matrix[0][a]*(matrix[1][((b+1)%3]*matrix[2][((b+2)%3]-
matrix[1][((b+2)%3]*matrix[2][((b+1)%3]));

}

if(d_matrix==0){

    cout<<"all three points are linear"<<endl;

    cout<<"Max distance between 2 points is: ";

    float max_distance=0;

    max_distance=sqrt(pow(matrix[0][0]-matrix[1][0],2)+pow(matrix[0][1]-
matrix[1][1],2));

    float temp=0;

    temp=sqrt(pow(matrix[0][0]-matrix[2][0],2)+pow(matrix[0][1]-
matrix[2][1],2));

    if(temp>max_distance){

        max_distance=temp;

        temp=0;

    }

    temp=sqrt(pow(matrix[1][0]-matrix[2][0],2)+pow(matrix[1][1]-
matrix[2][1],2));

    if(temp>max_distance){

        max_distance=temp;

        temp=0;

```

```

    }

    cout<<max_distance;
}

else{

    cout<<"all three points are not linear. They form a triangle."<<endl;

    cout<<"area of the triangle is: ";

    if(d_matrix < 0){

        d_matrix*=(-1);

    }

    cout<<d_matrix/2;

}

}

```

```

main.cpp
21     max_distance=temp;
22     temp=0;
23 }
24
25     temp=sqrt(pow(matrix[1][0]-matrix[2][0],2)+pow(matrix[1][1]-matrix[2][1],2));
26
27     if(temp>max_distance){
28         max_distance=temp;
29         temp=0;
30     }
31     cout<<max_distance;
32 }
33
34 else{
35     cout<<"all three points are not linear. They form a triangle."<<endl;
36     cout<<"area of the triangle is: ";
37     if(determinant_matrix < 0){
38         determinant_matrix*=(-1);
39     }
40     cout<<determinant_matrix/2;
41 }
42

```

input

```

all three points are not linear. They form a triangle.
area of the triangle is: 0.5
...Program finished with exit code 0
Press ENTER to exit console.

```

4).

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
void encrypt(string &s){  
    int length=s.length();  
    for(int i = 0 ; i<length ; i++){  
        if(i%2==0){  
            s[i]++;  
        }  
        else{  
            s[i]+=2;  
        }  
    }  
}
```

```
void decrypt(string &s){  
    int length=s.length();  
    for(int i = 0 ; i<length ; i++){  
        if(i%2==0){  
            s[i]--;  
        }  
        else{  
            s[i]-=2;  
        }  
    }  
}
```

The image shows a web browser window with the OnlineGDB C++ compiler. The browser's address bar shows the URL 'onlinegdb.com/online\_c++\_compiler'. The OnlineGDB interface includes a sidebar on the left with navigation links: 'IDE', 'My Projects', 'Classroom' (marked 'new'), 'Learn Programming', 'Programming Questions', 'Jobs' (marked 'new'), 'Sign Up', and 'Login'. Below these is a footer with links: 'About', 'FAQ', 'Blog', 'Terms of Use', 'Contact Us', 'GDB Tutorial', 'Credits', 'Privacy', and '© 2016 - 2023 GDB Online'. The main area has a top toolbar with buttons for 'Run', 'Debug', 'Stop', 'Share', 'Save', and 'Build'. Below the toolbar, the file 'main.cpp' is open, displaying a C++ program that implements a Caesar cipher. The code includes headers, namespace declarations, and functions for encrypting and decrypting a string. The 'encrypt' function shifts each character forward by 2 positions, while the 'decrypt' function shifts each character backward by 2 positions. At the bottom of the interface, there is an 'input' section with a text area containing the string 'my name is jeevan', a 'Run' button, and the output of the program, which shows the encrypted string 'n(!pbof"ju!lfgwco' and the decrypted string 'my name is jeevan'.

```
#include <iostream>
```

```
#include <ctime>
```

```
int getRandom(int min, int max) {  
    return min + (rand() % (max - min + 1));  
}
```

```
int main() {
    srand(time(0));
```



```
const int NUM_HORSES = 4;

const int POWER_INCREMENT = 5;

const int POWER_THRESHOLD = 10;


char horses[NUM_HORSES] = {'X', 'M', 'T', 'D'};

int power[NUM_HORSES] = {5, 5, 5, 5};

int steps[NUM_HORSES] = {0, 0, 0, 0};


bool raceOver = false;


for (int i = 0; i < NUM_HORSES; ++i) {
    cout << string(steps[i], '-') << horses[i] << string(20 - steps[i], '.') << "|" <<
endl;
}


cout << "Which horse do you think will win? (X, M, T, D): ";

char guess;

cin >> guess;


while (!raceOver) {
    for (int i = 0; i < NUM_HORSES; ++i) {
        power[i] += getRandom(1, 5);

        if (power[i] >= POWER_THRESHOLD * (steps[i] + 1)) {
            steps[i]++;
        }
    }
}
```

```

    }

    if (steps[i] >= 20) {
        raceOver = true;
    }
}

system("cls");

for (int i = 0; i < NUM_HORSES; ++i) {
    cout << string(steps[i], '-') << horses[i] << string(20 - steps[i], '.') << " | " <<
endl;
}

if (raceOver) {
    int winnerIndex = 0;
    for (int i = 1; i < NUM_HORSES; ++i) {
        if (steps[i] > steps[winnerIndex]) {
            winnerIndex = i;
        }
    }

    if (guess == horses[winnerIndex]) {
        cout << "Congratulations! You guessed the winner!" << endl;
    } else {

```

```

        cout << "Sorry, the winner was horse " << horses[winnerIndex] << endl;
    }

}

return 0;
}

```

The screenshot shows the OnlineGDB online C++ compiler interface. The code editor displays a C++ program with the following content:

```

main.cpp
15  const int POWER_THRESHOLD = 10;
16
17  char horses[NUM_HORSES] = {'X', 'M', 'T', 'D'};
18  int power[NUM_HORSES] = {5, 5, 5, 5};
19  int steps[NUM_HORSES] = {0, 0, 0, 0};
20
21  bool raceOver = false;
22
23  for (int i = 0; i < NUM_HORSES; ++i) {
24      cout << string(steps[i], '-') << horses[i] << string(20 - steps[i], '.') << "|" << endl;
25  }
26
27  cout << "Which horse do you think will win? (X, M, T, D): ";
28  char guess;

```

The output window shows the program's execution results:

```

-----X..|
-----M...|
-----T...|
-----D...|
sh: 1: cls: not found
-----X..|
-----M...|
-----T...|
-----D...|
sh: 1: cls: not found
-----X..|
-----M...|
-----T...|
-----D...|
Sorry, the winner was horse X
...Program finished with exit code 0

```

```

int main(){

    cout<<"enter a string that you want to encrypt"<<endl;

    string s;

    getline(cin,s);

    encrypt(s);

```

```
cout<<"encrypted string is :"<<endl;
```

```
cout<<s<<endl;
```

```
cout<<"for decrypting the string press 1 "<<endl;
```

```
int a;
```

```
cin>>a;
```

```
if(a==1){
```

```
    decrypt(s);
```

```
    cout<<s;
```

```
}
```

```
return 1;
```

```
}
```