

1. Write a program that takes a temperature value from the user. It should then allow the user to choose between Celsius (C) and Fahrenheit (F) for conversion. After the user selection, it should then convert the entered temperature to the chosen scale and display the result.

Use appropriate data types for temperature and handle error like non-numeric input.

Use the following formula for conversion:

$$F = (C \times 9/5) + 32$$

$$C = (F - 32) \times 5/9$$
 [10 marks]

**SOLUTION:**

```
#include <iostream>
```

```
using namespace std;
```

```
class temp
```

```
{
```

```
    private:
```

```
    float temp_value, result;
```

```
    int choice;
```

```
    public:
```

```
    void getdata()
```

```
    {
```

```
        cout<<"Enter the temperature"<<endl;
```

```
        cin>>temp_value;
```

```
    }
```

```
    void convert()
```

```
    {
```

```
        cout<<"will the temperature be Fahrenheit or Celsius(type '1' for Fahrenheit and '2' for Celsius)"<<endl;
```

```
        cin>>choice;
```

```
        if (choice==1)
```

```
        {
```

```
            result=(temp_value - 32)/1.8;
```

```
        }
```

```

        else if (choice==2)
        {
            result= (1.8 * temp_value)+32;
        }
        else
        {
            cout<<"Invalid input please enter '1' or '2'"<<endl;
        }
    }

    void display()
    {
        cout<<"The result after conversion will be "<<result<<endl;
    }

};

int main()
{
    temp t;
    t.getdata();
    t.convert();
    t.display();
    return 0;
}

```

**OUTPUT:**

```
"C:\Desktop\c++ assignment" X + v
Enter the temperature
23
will the temperature be Fahrenheit or Celsius(type '1' for Fahrenheit and '2' for Celsius)
2
The result after conversion will be 73.4

Process returned 0 (0x0)   execution time : 9.751 s
Press any key to continue.
```

```
"C:\Desktop\c++ assignment" X + v
Enter the temperature
73.4
will the temperature be Fahrenheit or Celsius(type '1' for Fahrenheit and '2' for Celsius)
1
The result after conversion will be 23

Process returned 0 (0x0)   execution time : 13.307 s
Press any key to continue.
```

2. Write a C++ program to implement a number guessing game with different difficulty levels.

Easy difficulty ranges from 1-8, medium from 1-30, hard from 1-50. Then, generate a random number to check if the guess is correct based on the user's selection.

[10 marks]

**SOLUTION:**

```
#include <iostream>
```

```
#include <cstdlib>
```

```
#include <ctime>
```

```
using namespace std;
```

```
class guess
```

```
{
```

```
private:
```

```
    int num;
```

```
    int choice;
```

```
    int random;
```

```
public:
```

```
    void level()
```

```
{
```

```

    cout<<"Enter which level do you want to play. \n"
    <<"Type '1' - Easy \n"
    <<"Type '2' - Medium \n"
    <<"Type '3' - Hard \n"<< endl;
    cin>>choice;
}

void display()
{
    if (choice==1)
    {
        int random = rand() % 8 + 1;
        cout<<"Guess the correct number within the range of 1-8"<<endl;
        cin>>num;
        if (num== random)
        {
            cout<<"The guess was correct! U WIN! "<<"The number is "<<num<<endl;
        }
        else
        {
            cout<<"The guess was not correct! U LOSE! "<<"The number is "<<random<<endl;

        }
    }
    else if (choice==2)
    {
        int random = rand()% 30 + 1;
        cout<<"Guess the correct number within the range of 1-30"<<endl;
        cin>>num;
        if (num== random)
        {

```

```

        cout<<"The guess was correct! U WIN! "<<"The number is "<<num<<endl;
    }
    else
    {
        cout<<"The guess was not correct! U LOSE! "<<"The number is "<<random<<endl;

    }
}

else if (choice==3)
{
    int random = rand() % 100 + 1;

    cout<<"Guess the correct number within the range of 1-100"<<endl;
    cin>>num;
    if (num== random)
    {
        cout<<"The guess was correct! U WIN! "<<"The number is "<<num<<endl;
    }
    else
    {
        cout<<"The guess was not correct! U LOSE! "<<"The number is "<<random<<endl;

    }
}

else
{
    cout<<"Invalid Input"<<endl;
}
}

};

int main()
{

```

```
    srand(time(0));

    guess g;

    g.level();

    g.display();

    return 0;

}
```

OUTPUT:

```
"C:\Desktop\c++ assignment" X + v
Enter which level do you want to play.
Type '1' - Easy
Type '2' - Medium
Type ' 3' - Hard

1
Guess the correct number within the range of 1-8
5
The guess was not correct! U LOSE! The number is 2

Process returned 0 (0x0)    execution time : 5.851 s
Press any key to continue.
```

```
"C:\Desktop\c++ assignment" X + v
Enter which level do you want to play.
Type '1' - Easy
Type '2' - Medium
Type ' 3' - Hard

2
Guess the correct number within the range of 1-30
23
The guess was not correct! U LOSE! The number is 15

Process returned 0 (0x0)    execution time : 11.513 s
Press any key to continue.
```

```
"C:\Desktop\c++ assignment" × + ∨
Enter which level do you want to play.
Type '1' - Easy
Type '2' - Medium
Type '3' - Hard

3
Guess the correct number within the range of 1-100
45
The guess was not correct! U LOSE! The number is 35

Process returned 0 (0x0)   execution time : 6.111 s
Press any key to continue.
```

3. Write a program that reads an array of integer numbers from the user and sorts the numbers in the ascending order.

[10 marks]

**SOLUTION:**

```
#include <iostream>
using namespace std;
int main()
{
    int n, hold;
    cout<<"Enter the amount of numbers you want to enter : "<<endl;
    cin>>n;
    int a[n];
    cout<<"Enter all the numbers: "<<endl;
    for (int i=0;i<n;i++)
    {
        cin>>a[i];
    }
    for (int i=0;i<n;i++)
    {
```

```

        for (int j= i+1; j<n; j++)
        {
            if (a[i]>a[j])
            {
                hold = a[i];
                a[i]=a[j];
                a[j]=hold;
            }
        }

    }

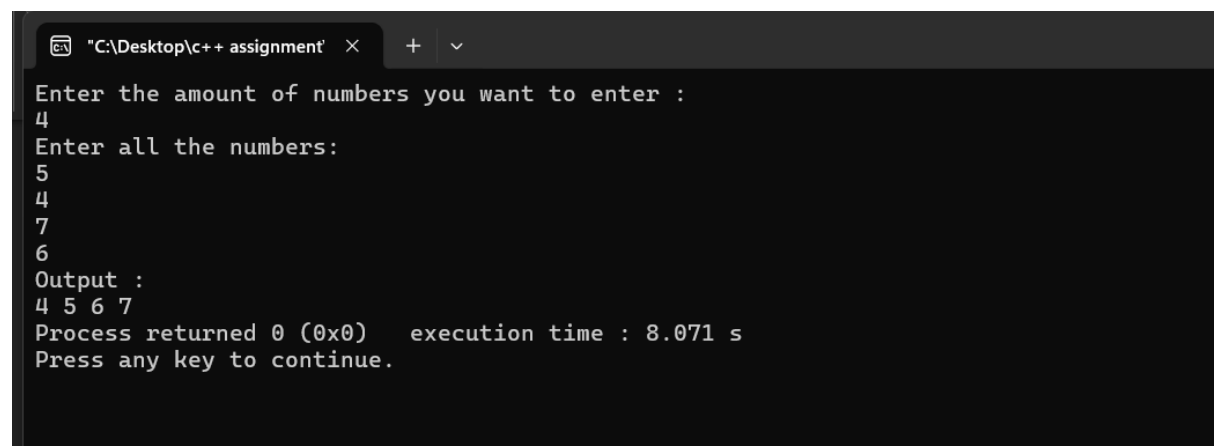
    cout<<"Output : "<<endl;
    for (int i=0; i<n; i++)
    {
        cout<<a[i]<<" ";
    }

    return 0;

}

```

**OUTPUT:**



```

"C:\Desktop\c++ assignment" x + v
Enter the amount of numbers you want to enter :
4
Enter all the numbers:
5
4
7
6
Output :
4 5 6 7
Process returned 0 (0x0)   execution time : 8.071 s
Press any key to continue.

```



4. Write a program that reads a number from the user and based on the user input, it says what day of the week it is, Sundays being 1 and Saturdays being 7. Your system should give appropriate response for invalid input entries. **[20 marks]**

**SOLUTION:**

```
#include <iostream>

using namespace std;

int main()
{
    int number;

    cout<<"enter any number from 1-7"<<endl;

    cin>>number;

    if (number==1)
    {
        cout<<"SUNDAY";
    }

    else if (number==2)
    {
        cout<<"MONDAY";
    }

    else if (number==3)
    {
        cout<<"TUESDAY";
    }

    else if (number==4)
    {
        cout<<"WEDNESDAY";
    }

    else if (number==5)
    {
        cout<<"THURSDAY";
    }
}
```

```

else if (number==6)
{
    cout<<"FRIDAY";
}
else if (number==7)
{
    cout<<"SATURDAY";
}
else
{
    cout<<"INVALID ENTER NUMBER BETWEEN 1-7";
}
return 0;
}

```

OUTPUT:

```

C:\Desktop\c++ assignment
enter anynumber from 1-7
3
TUESDAY
Process returned 0 (0x0)   execution time : 1.642 s
Press any key to continue.

```

## Task 2: Programming Exercises:[Control Statements]

1. Create a program that takes a positive integer as input and determines whether it's a "bouncy number". A bouncy number is one where the digits neither consistently increase nor consistently decrease when read from left to right. For example:

- 123 is NOT bouncy (digits consistently increase)
- 321 is NOT bouncy (digits consistently decrease)
- 120 is bouncy (neither consistently increasing nor decreasing)

[25 marks]

#### SOLUTION:

```
#include <iostream>

using namespace std;

int main()
{
    int num;
    int digit[10];

    cout<<"Enter a number which is atleast 3 digits and less then 10 digits"<<endl;
    cin>>num;

    if (num<100 || num>1000000000)
    {
        cout<<"INVALID INPUT PLEASE ENTER NUMBER OF 3-10 DIGITS";
    }

    //first extract the each digit of the number and put them into an array
    int i=0;
    while(num>0)
    {
        digit[i]= num%10;
        num = num / 10;
        i++;
    }

    //numbers stored in reverse order
    //read the array in correct order
    bool up = true, down = true;
    for (int j=i-1;j>0;j--)
    {
        //set condition to check if bouncy
        if (digit[j]>digit[j-1])
        {

```

```
        up=false;
    }
    else if (digit[j]<digit[j-1])
    {
        down=false;
    }
}
```

```
if (!up && !down)
```

```
    cout << "The number is BOUNCY." << endl;
```

```
else if (up)
```

```
    cout << "The number is increasing." << endl;
```

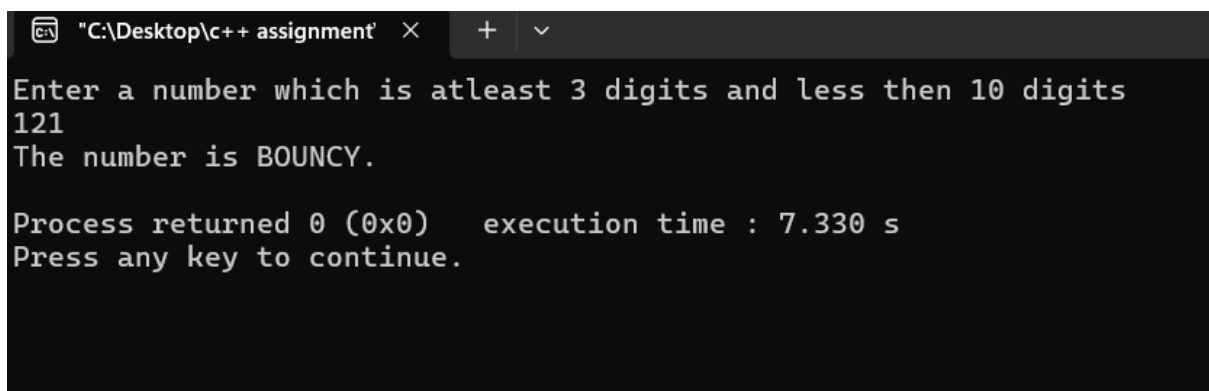
```
else if (down)
```

```
    cout << "The number is decreasing." << endl;
```

```
return 0;
```

```
}
```

OUTPUT:



```
"C:\Desktop\c++ assignment" X + v
Enter a number which is atleast 3 digits and less then 10 digits
121
The number is BOUNCY.

Process returned 0 (0x0)   execution time : 7.330 s
Press any key to continue.
```

```
"C:\Desktop\c++ assignment" X + v
Enter a number which is atleast 3 digits and less then 10 digits
123
The number is increasing.

Process returned 0 (0x0)   execution time : 1.827 s
Press any key to continue.
```

```
"C:\Desktop\c++ assignment" X + v
Enter a number which is atleast 3 digits and less then 10 digits
321
The number is decreasing.

Process returned 0 (0x0)   execution time : 2.003 s
Press any key to continue.
```

### Task 3: Programming Exercises on Arrays

1. Write a program that manages a cinema ticket booking system. The program should display a 5x5 seating arrangement where:

1. Available seats are marked with 'O'
2. Booked seats are marked with 'X'

Program should:

1. Display the current seating arrangement
2. Ask user for row and column number (1-5) for booking
3. Mark that seat as booked ('X')
4. Show updated seating after each booking
5. Display error if user selects already booked seat
6. Display error if user enters invalid row/column numbers

**SOLUTION:**

```
#include <iostream>
```

```
using namespace std;
```

```

class Cinemabook
{
private:
    int row,column;
    char seat[5][5];
public:
    //set the seats as "O"
    void defaultseat()
    {
        for (int i=0;i<5;i++)
        {
            for (int j=0;j<5;j++)
            {
                seat[i][j] = 'O';
            }
        }
    }
    //ask which seat the user wants to book
    void booking()
    {
        cout<<"Enter the row"<<endl;
        cin>>row;
        cout<<"Enter the column"<<endl;
        cin>>column;
    }
    // set conditions
    void bookseats()
    {
        if (0>row>5 || 0>column>5)
        {
            cout<<"enter a valid row and column !"<<endl;

```

```

    }
    else if (seat[row-1][column-1]=='X')
    {
        cout << "Seat already booked pick another one."<<endl;
    }
    else
    {
        seat[row-1][column-1] = 'X';
    }
}

//display the seating arrangement after booking
void finalseating()
{
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            cout << seat[i][j] << " ";
        }
        cout << endl;
    }
}

};

```

```

int main(){
    Cinemabook c1;
    c1.defaultseat();

```

```

    int loop;

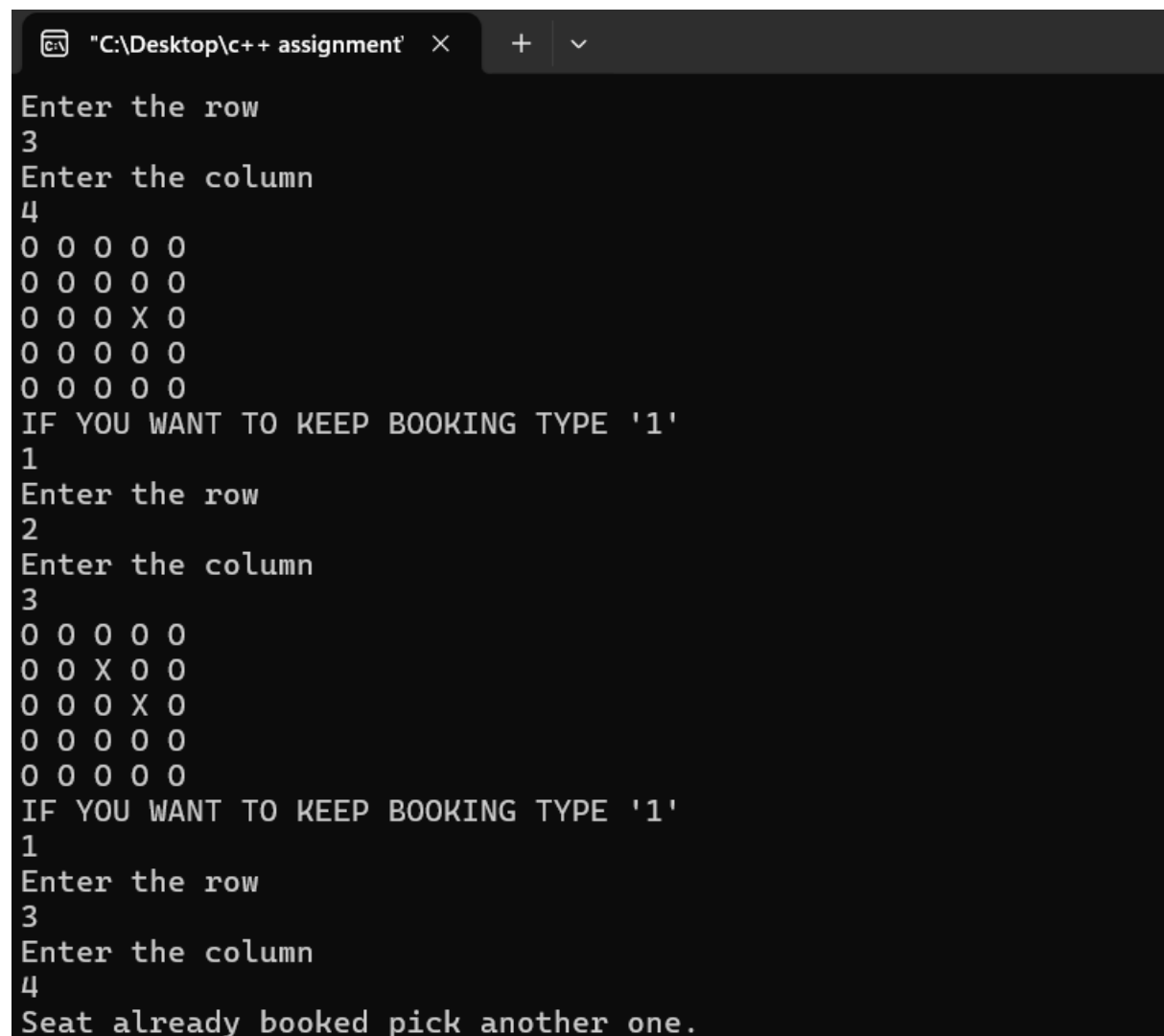
```

```

do
{
c1.booking();
c1.bookseats();
c1.finalseating();
cout<<"IF YOU WANT TO KEEP BOOKING TYPE '1'"<<endl;
cin>>loop;
} while (loop==1);
return 0;
}

```

OUTPUT:



```

C:\Desktop\c++ assignment
Enter the row
3
Enter the column
4
0 0 0 0 0
0 0 0 0 0
0 0 0 X 0
0 0 0 0 0
0 0 0 0 0
IF YOU WANT TO KEEP BOOKING TYPE '1'
1
Enter the row
2
Enter the column
3
0 0 0 0 0
0 0 X 0 0
0 0 0 X 0
0 0 0 0 0
0 0 0 0 0
IF YOU WANT TO KEEP BOOKING TYPE '1'
1
Enter the row
3
Enter the column
4
Seat already booked pick another one.

```



