WEEK:: 02





COURSE INSTRUCTOR BY ROHIT NEGI

MADE BY-PRADUM SINGHA

<u>Check Profile</u> <u>My Profile</u>

WEEK :: 02 DAY: 01 DATE: - 24-04-2023

BINARY TO DECIMAL + FUNCTIONS

Binary	Decimal
1 0 1 0	10

Convert ::

1	1 0 1		0
1X2^3	1X2^2	1X2^1	0X2^0
8	0	2	0

To Convert Binary divided by 2 because there are 2 digits 0 and 1.

To Convert Decimal divided by 10 because there are 10 digits 0 to 9.

Binary to Decimal::

1 0 1 0 first take last digit	1010 %10 ⇒0 X 2^0	cin>>num, sum = 0; int mul = 1;
1 0 1	101 %10 ⇒1 X 2^1	while(num>0)
1 0	10 %10 ⇒0 X 2^2	ไnt rem= num%10; num = num%10;
1	1 %10 ⇒1 X 2^3	sum = sum+rem X mul; mul = mul X 2; } cout< <sum;< td=""></sum;<>

#Code::

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

int main()
{
   int num, sum =0, mul = 1;
   cout<<"Enter Binary Digit: ";
   cin>>num;
```

```
while(num>0)
{
    int rem= num%10;
    num = num/10;
    sum = sum+ rem*mul;
    mul = mul*2;
}
cout<<sum;

return 0;
};</pre>
```

Use a long long Data Type when entering a long input like (101011110001110).

1-s Complement:

[Details know String lecture]

#Scope ::

code	Global Variable	Local Variable
int sum; int mul=0; Int main()	Here sum, mul is a Global Variable because any where can exceed .	Here rem is aLocal Variable because rem only excesses inside { }.
{ Int rem = ;		Whatever variable is declared inside of { } It is included in the Local Variable.
}		
cout< <sum;< td=""><td></td><td></td></sum;<>		

#Break and Continue::

Break :: It is used to terminate the loop.

Continue:: It is used to execute the next iteration in a particular condition.

Break Continue #include<iostream> #include<iostream> using namespace std; using namespace std; int main() int main() int num; int num; for(int i=1; i<=10; i++)</pre> for(int i=1; i<=10; i++)</pre> { **if**(i==6) **if**(i==6) cout<<i<" "; cout<<i<" "; } } **}**;

SWITCH CONDITION

```
Syntex

switch(expression)
{
    case 1:
        cout<<" .....";
        case 1:
        cout<<" .....";
        default :
        cout<<".....";
}
```

#Code

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

int main()
{
    int num;
    cout<<"Enter the Num: ";
    cin>>num;

    switch(num)
    {
        case 1: cout<<"Sunday";
        break;
        case 2: cout<<"Munday";
        break;
        case 3: cout<<"Tuesday";</pre>
```

```
break;
    case 4: cout<<"Wednesday";
    break;
    case 5: cout<<"Thursday";
    break;
    case 6: cout<<"Friday";
    break;
    case 7: cout<<"Saturday";
    break;
    default: cout<<"Number not valid";
}

return 0;
};</pre>
```

FUNCTION

A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function.

```
include <iostream>
                                                 int isEvenOdd (int num)
sing namespace std;
                                                  #code- Function Define
int isEvenOdd(int num)
                                                int - return int value
                                                Int num - int parameter
                                                void - no return value
                                                Default Argument:
int main()
                                                int isEvenOdd(int num =5)
                                                {
                                                }
    cout << "Enter Num: ";</pre>
                                                isEvenOdd(num) - call Function
    int n = isEvenOdd(num); // call
                                                (num) - Argument
        cout << "Even";</pre>
                                                Syntex ::
                                                          return_type Fun_name (Parameter)
        cout << "Odd";
                                                             // code
    isEvenOdd(num);
                                                          }
```

Exp:: 01: Print Factorial

#include <iostream>

```
using namespace std;

// create function
int printFactorial(int num)
{
   int result = 1;
   for(int i=1; i<=num; i++)
   result = result*i;

   return result;
}

int main()
{
   int num;
   cout<<"Enter Num: ";
   cin>>num;

   int ans =printFactorial(num);
   cout<<ans;
   return 0;
};</pre>
```

When passing big values like 18, 20, 123, use (long long) data type.

HOMEWORK

Exp:: 01 Calculate Average value of Two numbers using Function.

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

// create function
int avgTwoNum (int num1, int num2)
{
    int sum = num1+num2;
    int avg = sum/2;
    return avg;
}

int main()
{
    int num1, num2;
    cout<<"Enter Num1: ";
    cin>>num1;
    cout<<"Enter Num2: ";
    cin>>num2;
    int result =avgTwoNum(num1, num2);
    cout<<result;
    return 0;
};</pre>
```

Exp::02 : Find Prime Number.

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

void findPrime(int num)
{
   if (num < 2)
    {
       cout<<"Not Prime";
   }
   for (int i=2; i<num; i++)
   {
       if (num%i ==0)
      {
            cout<<"Not Prime";
            return;
        }
       else</pre>
```

```
cout<<"Prime";
    return;
}

int main()
{
    int num;
    cout<<"Enter Num: ";
    cin>>num;
    findPrime(num);

    return 0;
};
```

Exp:: 03 Print Fibonacci Series ::

WEEK :: 02 DAY: 02 DATE: - 25-04-2023

FUNCTIONS + ARRAY

Exp :: 01 :: Print "Odd" and "Even".

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

void IsEvenOdd(int num)
{
    if (num%2==0)
        cout<<"Even"<<endl;
    else
        cout<<"Odd"<<endl;
    return;
}
int main()
{
    int num1, num2, num3;
        cin>>num1>>num2>>num3;
        IsEvenOdd(num1);
        IsEvenOdd(num2);
        IsEvenOdd(num3);
return 0;
};
```

Exp :: 02 :: Add Two numbers 8 and 4.

```
#include<iostream>
using namespace std;
void Add()
{
    cout<<8+4;
}
int main()
{
    Add();</pre>
```

```
return 0;
};
```

Exp :: 03 :: Add Two numbers take input User?

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

int AddTwoNum(int num1, int num2)
{
    int sum = num1+num2;
    return sum;
}

int main()
{
    int num1, num2;
    cin>num1>>num2;
    int result = AddTwoNum(num1,num2);
    cout<<result;

return 0;
};</pre>
```

Exp :: 03 :: Print Table ?

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

void PrintTable(int n)
{
    for(int i=1; i<=10; i++)
        cout<<n*i<<endl;
    return;
}
int main()
{
    int n;
    cin>>n;
    PrintTable(n);
return 0;
};
```

Exp :: 03 :: Print Factorial?

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

int PrintFactorial(int num)
{
    int total = 1;
    for(int i=1; i<=num; i++)
    total = total*i;
    return total;
}

int main()
{
    int num;
    cin>>num;
    int result = PrintFactorial(num);
    cout<<result;
return 0;
};</pre>
```

FUNCTION OVERLOADING

Function overloading or method overloading is the ability to create multiple functions of the same name with different implementations.

```
#include<iostream>
using namespace std;
void Add(int num1, int num2)
{
    int sum = num1 + num2;
    cout<<sum<<" ";
    return;
}
void Add(int num1, int num2, int num3)
{
    int sum = num1 + num2 + num3;
    cout<<sum<<" ";
    return;
}
int main()
{
    Add(20, 30);</pre>
```

```
Add(10, 10, 10);
return 0;
};
```

Function execute according to Argument and Parameter (Argument = Parameter).

Call by value:: When pass value using Argument.

ARRAY

Arrays are used to store multiple values in a single variable with the same data type. Define ::

int arr[i];

7	6	8	8	4	6
Index: 0	1	2	3	4	5

Define Array::

$$arr[6] = \{7, 6, 8, 8, 4, 6\};$$

Excese Array ::

```
cout<<arr[i];
cout<<arr[0];
cout<<arr[2];
cout<<arr[3];
cout<<arr[4];</pre>
```

WEEK :: 02 DAY: 03 DATE: - 26-04-2023

Basic Questions in Array

Array Position ::

Arr[i] = Base address + i*size of Variable

Exp::01 :: print Arr reverse ::

```
#include<iostream>
```

```
using namespace std;
int main()
{
    int arr[5] = {6, 9, 7, 3, 1};
    for(int i=4; i>=0; i--)
        cout<<arr[i]<<" ";

return 0;
};</pre>
```

Exp::02:: Find Largest numbers in Array?

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

int main()
{
    int arr[6] = {6, 2, 4, 9, 7, 8};
    int largest = arr[0];
    for(int i=1; i<6; i++)
    {
        if(arr[i]>largest)
        largest = arr[i];
    }
    cout<<largest;

return 0;
};</pre>
```

Exp::03 :: Find Odd numbers in Array?

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

int main()
{
   int arr[8] = {2, 5, 3, -3, 8, 4, 2, -6};
   for(int i=0; i<=8; i++)
   {
      if(arr[i] %2==1 || arr[i] %2==-1)
      cout<<arr[i]<<" ";
   }
return 0;
};</pre>
```

Exp::05 :: Print prime numbers in Array? (Using Function)

```
include <iostream>
using namespace std;
   cout << num << " ";
      prime(arr[i]);
```

Exp::06 :: Print Array input: {2, 3, 7, -11, 4} Output: {4, 2, 3, 7, 11}

```
finclude<iostream>
using namespace std;

int main()
{
    int arr[5] = {2, 3, 7, -11, 4};
    int last_arr = arr[4];

    for(int i=3; i>=0; i--)
    {
        arr[i+1]=arr[i];
    }
    arr[0]=last_arr;

    for(int i = 0; i<5; i++)
        cout<<arr[i]<<" ";

return 0;
};</pre>
```

Exp::07 :: Print Array [-No repeat element] input: {2, 4, 6, 4, 2, 8, 6} Output:{ 8 }

HOMEWORK

Exp::08 :: Print smallest number in array?

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

int main()
{
    int arr[8] = {7, 11, -2, 8, 170, -4, 0, 11};
    int small_N = arr[0];

    for(int i=1; i<=7; i++)
    {
        if(small_N>arr[i])
        {
            small_N=arr[i];
        }
    }
    cout<<small_N;

return 0;
};</pre>
```

Exp::09 :: Print Even number in array?

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

int main()
{
    int arr[8] = {2, 3, 7, -6, -11, 8, 13, 12};
    for(int i=0; i<8; i++)
    {
        if(arr[i]*2==0)
        {
            cout<<arr[i]<<" ";
        }
    }

return 0;
};</pre>
```

Exp::09:: Print Prime number in array? (WithOut Function)

Try Yourself



WEEK :: 02 DAY: 04 DATE: - 27-04-2023

Print Array Element + Sorting Algorithm

Exp::01 :: Print array 5 times?

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

int main()
{
    int arr[5] = {2, 3, 8, 7, 4};

    for (int i = 0; i < 5; i++)
        {
        for (int j = 0; j < 5; j++)
            cout << arr[j] << " ";

        cout << endl;
    }

    return 0;
};</pre>
```

Exp::02 :: Print array 2 3 8 7 4 2 3 8 7 2 3 8 2 3 2

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

int main()
{
    int arr[5] = {2, 3, 8, 7, 4};

    for (int i = 0; i < 5; i++)
        {
        for (int j = 0; j < 5-i; j++)
            cout << arr[j] << " ";
        cout << endl;</pre>
```

```
Exp::03 :: Print array
3 13 12 7 4
13 12 7
12 7 4
7 4
using namespace std;
        cout<<arr[j]<<" ";
         cout<<endl;</pre>
```

Exp::04 :: Print array reverse ? Input: { 3, 11, 13, 8, 4}; **Output:** {4, 8, 13, 11, 3};

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

int main()
{
    int arr[5] = {3, 11, 13, 8, 4};
    for (int i = 0; i < 5; i++)
    {
        for (int j = 4; j >= 0; j--)
            cout << arr[j] << " ";

        cout << endl;
    }

    return 0;
};</pre>
```

```
Exp::05 :: Print array Input: {3, 13,12, 8, 4};
Output:
4 8 12 13 3
8 12 13 3
12 13 3
13 3
3
#include<iostream>
    int arr[5] = {3, 13, 12, 8, 4};
        cout<<arr[j]<<" ";
        cout<<endl;
Exp::06 :: Print array Input: {3, 13,12, 8, 4};
Output:
4 8 12 13 3
4 8 12 13
4812
48
4
#include<iostream>
using namespace std;
int main()
    int arr[5]={3, 13,12, 8, 4};
    for(int i=0; i<5; i++)</pre>
    {
        for(int j=4; j>=i;j--)
        cout<<arr[j]<<" ";
        cout<<endl;</pre>
```

Exp::07 :: Print Duplicate array?

```
#include<iostream>
using namespace std;
int main()
{
   int arr[5] = {2, 2, 3, 3, 4};
   int count[5]={0};

   for(int i=0; i<5; i++)
   {
      count[arr[i]]++;
   }
   for(int i=0; i<5; i++)
   {
      if(count[i]>1)
      cout<<ii<<" ";
   }

return 0;
};</pre>
```

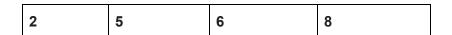
SORTING

sorting algorithm is an algorithm that puts elements of a list into an order.

Exp::

5 2 8 6

Sorting Array:



Exp::01 :: Print Sorted array? Input: {1, 0, 2, 1, 0} Output: {0, 0, 1, 1, 2}

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

int main()
{
    int arr[5] = {1, 0, 2, 1, 0};
    int count_zero = 0, count_one = 0, count_two = 0;
    for (int i = 0; i < 5; i++)
    {
        if (arr[i] == 0)
            count_zero++;
        else if (arr[i] == 1)</pre>
```

WEEK :: 02 DAY: 05 DATE: - 28-04-2023

SORTING ALGORITHM (Selection, Bubble, Insertion)

#Sorting in order:

8	6	7	2	4

Ascending Order:

2 4	6	7	8
-----	---	---	---

Descending Order:

|--|

We can easily find any item/numbers using the sorting Algorithm.

Arrange Ascending Order: Selection Sort

Input :: { 2, 7, 8, 3, 5, 1}; **Output**:: {1, 2, 3, 5, 7, 8};

Explain:

		2	7	8	3	5	1
--	--	---	---	---	---	---	---

Index	0	1	2	3	4	5

Round 1	0	1	7	8	3	5	2	Check 0-5
Round 2	1	1	2	8	3	5	7	Check 1-5
Round 3	2	1	2	3	8	5	7	Check 2-5
Round 4	3	1	2	3	5	8	7	Check 3-5
Round 5	4	1	2	3	5	7	8	Check 4-5

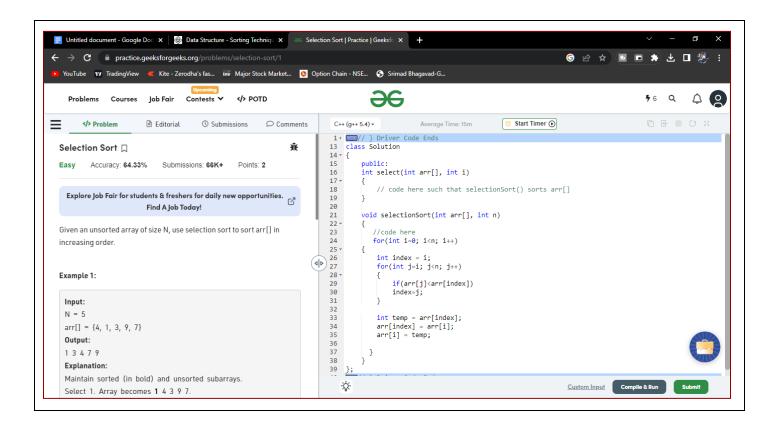
Swap::

Exchange	4	1
index	0	4

```
Int temp = arr[4]
arr[4] = arr[0]
arr[0] = temp;
```

Code::

GeeksforGeeks



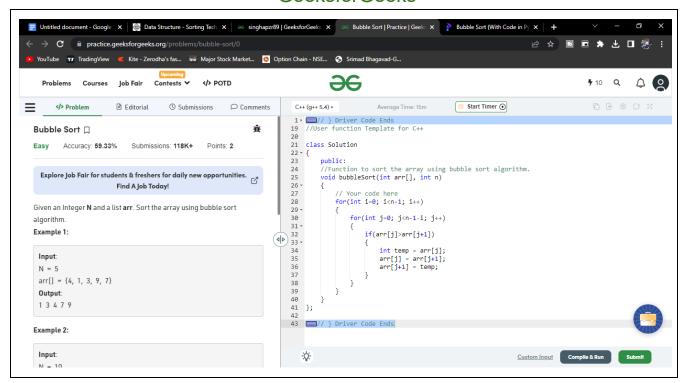
Bubble Sort

5 3 6 1 2	
-----------	--

Round 1	5	3	6	1	2	Check [(n-1)-i]
Round 2	3	5	1	2	6	[(n-1)-i]
Round 3	3	1	2	5	6	[(n-1)-i]
Round 4	1	2	3	5	6	[(n-1)-i]
Index	0	2	3	4	5	

```
include <iostream>
int main()
           if (arr[j] > arr[j + 1])
               int temp = arr[j];
              arr[j] = arr[j + 1];
               arr[j + 1] = temp;
   cout<<arr[i]<<" ";
```

GeeksforGeeks



Insertion Sort

4 6 3	11	7	2
-------	----	---	---

Round 1	4	6	3	11	7	2	Check 1-0
Round 2	3	4	6	11	7	2	2 - 0
Round 3	3	4	6	7	2	11	3 - 0
Round 4	3	4	6	2	7	11	4 - 0
Round 5	2	3	4	6	7	11	5- 0

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

int main()
{
    int arr[6] = {4, 6, 3, 11, 7, 2};

    for(int i=0; i<6-1; i++)
    {
        for(int j=i; j>=0; j--)
        {
            if(arr[j]>arr[j+1])
```

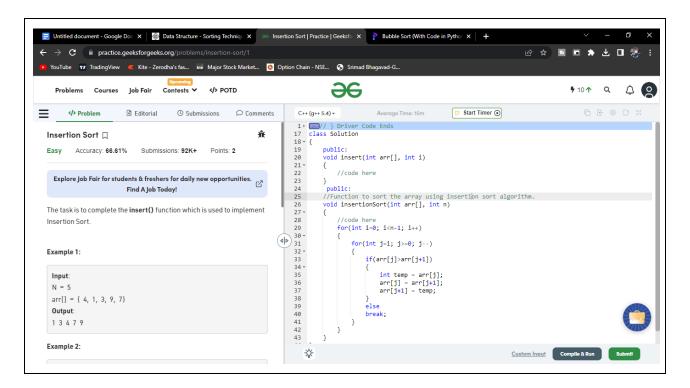
```
{
    int temp = arr[j];
    arr[j]= arr[j+1];
    arr[j+1] = temp;
}
else
break;
}
};

// print array

for (int i = 0; i < 6; i++)
    cout << arr[i] << " ";

return 0;
};</pre>
```

GeeksforGeeks



GeeksforGeeks

Find the fine::

```
long long int collection = 0;

/*

if(date%2==1)
{

for(int i = 0; i<n; i++)</pre>
```

```
{
    if(car[i]%2==0)
    {
        collection = collection + fine[i];
    }
}

if(date%2==0)
{
    for(int i=0; i<n; i++)
    {
        if(car[i]%2==1)
        {
        collection = collection + fine[i];
        }
    }
} */

for(int i=0; i<n; i++)
{
    if(date%2 != car[i]%2)
    collection = collection + fine[i];
}
</pre>
```

return collection;