Function And Recrusion:-

- Function is an individual set of instruction invokeable for a single purpose. Function allow to structure programs in segments of code to perform individual task. In C++ a function is a group of statement that is given a name and which can be called from some point of the program.

- We make/define function outside the main and recall it into main function.

- Type function name (Parameter list)

SQUARE FUNCTION:-

```
# include <io stream>
using namespace std;

// making function of square

int square(int num1) → function with one parameter

{ int squared;
squared = num1 * num1;
Return squared;

}

int main( )

{
// variable declaration
int a;

// input phase
cout << " Enter any integer";
cin >> a;
```

// Processing BY recalling function

```
cout << "The square of " << a
<< " is " << square(a);

Return 0;

}
```

OUTPUT:-

Enter any integer ____

The square of __ is ____

ALTERNATE SQUARE FUNCTION

```
# include <io stream>
using name space std;

// making function of square
int square (int num1 ) → function with
                          one parameter
{
Relūrn num1 * num1 ;
}

int main( )

{
// variable decleration
int a;

// input phase
cout << " Enter any integer ";
cin >> a ;

// Processing BY recalling function
```

```
cout << " The square of " << a << " is "
<< square(a);

Return 0;

}
```

OUTPUT:-

Enter any integer ___

The square of ___ is ___

• MAXIMUM NUMBER

FUNCTION:-

```
#include <iostream>
using namespace std;
//making function for largest number
int maximum(int num1, int num2, int num3)
                    └→ 3 parameters
{
 int largest;

 If (a>b)

 largest = a;

 else
 largest = b;

 If (c> largest)

 largest = c;

 Relūrn largest;

}
```

```
int main( )

{
// variable decleration
int a,b,c;

// input phase
cout << "Enter three integers";
cin >> a >> b >> c;

// processing by recalling function

cout << "The maximum among
 these three numbers is"
<< maximum (a,b,c) ;

return 0;

}
```


Scanned with CamScanner

OUTPUT:-

Enter three integers _ _ _

The maximum among these three numbers is ___

PREDICATE FUNCTIONS

Distinguist even and odd numbers.

```
# include <io stream>
using namespace std;

bool iseven(int num)

{ If (num%2 ==0)

   Relurn true ;

 else

Relurn false;

}

int main( )

{
// variable decleration
int num1;
```

```
//input phase
cout << "Enter any integer";
cin >> num1;
//Processing by recalling function.
if (isEven(num1))

{
cout << "Num" << num1 << "is even
number";

}

else

{
cout << "Num" << num1 << "is odd
number";

}

Return 0;

}
```


Scanned with CamScanner

OUTPUT:-

Enter any integer 2

Num 2 is even number

Scanned with CamScanner

Predicate Function To Check Vowel:-

```
# include <io stream>
using namespace std;

// make predicate function to check
 vowel
bool isvowel (char letter)

{
If (letter == 'a' || letter == 'e' ||
    letter == 'i' || letter == 'o' ||
    letter == 'u' )

Return true;

else

Return false;

}
```


Scanned with CamScanner

```
int main()
{
// variable declaration
char ltr;

// input phase
cout << "Enter any alphabet in
        small case";
cin >> ltr;

// processing by recalling function.
if (isvowel(ltr))
{
cout << "letter" << ltr << "is vowel";
}
else
{
cout << "letter" << ltr << "is
 consonant";
```


Scanned with CamScanner

```
}
Return 0;
}
```

OUTPUT:-

Enter any alphabet in small case e

letter e is vowel

Scanned with CamScanner

Void Function :-

Void function is for printing patterns in program.

// make any void function

```
# include <io stream>
using namespace std;

void heading ( )

{
cout << "**********************";
cout << "\n* EE-145 Compoter & Programming *" << endl;
cout << "* Electrical Engineering*";
cout << "\n**********************";

}

int main ( )

{
```


Scanned with CamScanner

// Recalling void function

```
heading ( );

return 0;

}
```

OUTPUT:-

```
**************************
* EE-145 Computer & programming *
* Electrical Engineering *
**************************
```


Scanned with CamScanner

Another Void Function :-

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

// making void function
void heading(string course, string batch)

{
cout << "**********\n";
cout << "**** " << course << " ****\n";
cout << "**** " << batch << " ****\n";
cout << "**********";

}

int main()

{
// Recalling void function

heading("CP", "FE - Electrical");
return 0;

}
```


Scanned with CamScanner

OUTPUT:-

```
**********************
****  CP  ****
***  FE-Electrical ****
**********************
```

// To convert given temperature in °C or °F

Function Call From Other Than Main:-

```
# include <io stream>
using namespace std;

double ctof (double temp)

{
return (temp * (9.0 / 5.0)) + 32 ;
}

double ftoc (double temp)

{
return (temp - 32.0) * 5.0 / 9.0 ;
}
```

```
double temperature converter (char scale,
    double temp)

{
if (char == 'C' || char == 'C')

{
cout << "Converted temperature from
  F to C is";
return ftoc (temp);
 }

if (char == 'f' || char == 'F')

{
cout << "converted temperature from
 C to F is";
 return ctof (temp);

 }

  else
{
 cout << "scale is incorrect";
 }
```


Scanned with CamScanner

```
}

int main( )

{
// variable declaration
double temperature;
char scaling;

// input phase

cout << "Enter temperature to convert";
cin >> temperature;

cout << "Enter scale in which You
 want to convert";
cin >> scaling;

// processing by recalling function

cout << "The converted temperature
is" << temperature converter(scaling,
 temperature);

Return 0;

}
```

OUTPUT:-

Enter temperature to convert ___

Enter scale in which you want to convert C

Converted temperature from F to C is _

SWAPPING NUMBER

WITHOUT FUNCTION:-

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

int main( )

{
// variable decleration
int num1 , num2 , temp;

// variable intialization
num1 = 10;
num2 = 13;

// printing numbers
cout << "num1 = " << num1 << endl;
cout << "num2 = " << num2 << endl;

// swapping number without function

temp = num2;
num1 = num2;
num2 = num1;
num1 = temp;
```


Scanned with CamScanner

// printing swapped numbers.

```
cout << "num1 = " << num1;
cout << "\n num2 = " << num2;

return 0;

}
```

OUTPUT:-

```
num1 = 10
num2 = 13

num1 = 13
num2 = 10
```


Scanned with CamScanner

SWAPPING NUMBER BY USING FUNCTION

```
# include <io stream>
using namespace std;

void swap(int num1, int num2)

{
  int temp;
temp = num2;
num2 = num1;
num1 = temp;

}

int main( )

{
 // variable dederation
 int a, b;

 // variable intialization
 a = 10;
 b = 13;
```

// printing numbers

```
cout << "num1 = " << a << endl;
cout << "num2 = " << b << endl;
```

// Recalling swap function:

```
swap(a,b);

cout << "num1 = " << a << endl;
cout << "num2 = " << b << endl;

return 0;

}
```

OUTPUT:-

```
num1 = 10
num2 = 13

num1 = 13
num2 = 10
```


Scanned with CamScanner

- Bubble sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order.

- **STEP: 01**

Starting with the first element (index=0). compare the current element with next element.

- **STEP: 02**

If the current element is greater than the next element of the array, we have to swap them.

Scanned with CamScanner

STEP: 03

If the current element is less than the next element, move to the next element

STEP: 04

Repeat step: 01 till the list is sorted.

Scanned with CamScanner

Bubble sort Algorithm

n: size of list
↳ SIZE

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

{ // swapping
int temp = arr[J];
 arr[J] = arr[J+1] ;
arr[J+1] = temp ;
 }

 }
```


Scanned with CamScanner

Q.24

Bubble-Sort Function

```
# include <io stream>
using name space std;

// making bubble_sort function

void bubble_sort (double arr[ ], int SIZE)

{
// first taking unsorted sequence
 in an array

cout << "\t UNSORTED SEQUENCE \n";

// using for-loop to take input
  in array

for (int i=0; i<SIZE; i++)

{
cout << "Element no. " << i << " = ";
cin >> arr[i];

}
```


Scanned with CamScanner

```
// Now apply bubble sort
      algorithm

// swapping numbers

for (int i=0; i<SIZE; i++)

{
for (int j=0; j<(SIZE-i-1); j++)

{
if (arr[j] > arr[j+1])

{
int temp = arr[j];
arr[j] = arr[j+1];
arr[j+1] = temp;

}

}

}
```


Scanned with CamScanner

```
// Now display sorted
      sequence

cout << endl;

cout << "\t SORTED SEQUENCE \n";

// for-loop to display sorted sequence
for (int i=0; i<SIZE; i++)
{
cout << "\n Element no." << i << "="
   << arr[i] <<endl;

 }

 }

# define element 10

 int main( )

{
```


Scanned with CamScanner

```
// declaring array
double myarray [element];
// Now recalling bubble-sort function:-
bubble_sort (myarray, element);
return 0;
}
```

Q.26

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

// insert_element function

void insert_element ( int arr [ ], int SIZE)

{
// for-loop to take input from
      user
for (int i=0; i<SIZE; i++)

{
cout << " Enter element no: " << i
 << " = " ;
cin>> arr [i];
cout <<endl;

}
cout <<endl <<endl;

}
```


Scanned with CamScanner

```
// index_element_replace function

void index_element_replace (int arr[ ], int
                                    SIZE)
{
 int index;
cout << "Enter index you want to
      replace: ";
cin >> index;

cout << "Enter new value for this
 index: ";

cin >> arr [index ];

cout<<endl <<endl;

 }
```


Scanned with CamScanner

```
// index_find_replace function

void index_find_replace (int arr[], int SIZE)

{
 int element;
cout << "Enter element you want to
  replace: ";
cin >> element;

for (int i=0; i<SIZE; i++)

{
If (element == arr[i])

{
cout << "Your element" << element << "found
at" << i << "position" << endl;

cout << "Enter new value for this
  index: ";

cin >> arr[i];
}
}
cout << endl << endl;
```


Scanned with CamScanner

```
}

// index_element_remove function

void index_element_remove(int arr[],
                          int SIZE)
{
  int index;
  cout << "Enter index you want to
          remove: ";
cin >> index;

for (int i=index; i<SIZE; i++)
 {
int temp = arr[i];
 arr[i] = arr[i+1];
 arr[i+1] = temp;

 }

}
cout<<endl;
```


Scanned with CamScanner

```
// find element

void find_element (int arr[ ], int SIZE)

{
 int element;
 cout << "Enter element You want to
 find :";

 cin>> element;

 for (int i=0; i<SIZE; i++)

 {
 If (element == arr[i])

 {
 cout << " Your element found at
 index no. " << i << endl;

 }

 }

 cout<<endl;
   }
```


Scanned with CamScanner

```
// display function

void display (int arr[]; int SIZE)

{
for (int i=0; i<SIZE; i++)

{
cout << " Element no. " << i << " = "
 << arr[i] << endl;

 }
cout<<endl;

 }

# Define size 10

int main( )

{
// declaring array
int my_array [size];
```


Scanned with CamScanner

```
// Recalling insert_element function
insert_element (myarray, size);

// Recalling display function
display (my array, size);

// Recalling index_find_replace function
index_find_replace (my array, size);

// Recalling display function
display (my array, size);

// Recalling index_element_replace function
index_element_replace (myarray, size);

// Recalling display function
display (myarray, size);

// Recalling find_element function
find_element (myarray, size);

// Recalling index_element_remove function
index_element_remove (myarray, size);
```


Scanned with CamScanner

```
// Recalling display function

display (myarray, size-1);

return 0;

}
```


Scanned with CamScanner