**Q-1.** Write a function that takes as input a number and outputs its square.

**Ans:**

#include <stdio.h>

int getSquare(int x);

void main()

{

int a;

scanf("%d", &a);

int ans = getSquare(a);

printf("Square of %d is %d\n", a, ans);

}

int getSquare(int n)

{

int s = n\*n;

return s;

}

**Q-2.** How do you declare a function that takes as input an integer and a double and returns a char?

**Q-3.** Suppose you have a function gcd(a, b)that finds the gcd of two numbers. Write a program that takes as input 10 integers and finds their gcd.

**Ans:**

#include <stdio.h>

int getGCD(int p, int q);

int getGcdMulti(int n, int value[]);

void main()

{

int n=10, value[10];

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

{

scanf("%d", &value[i]);

}

int ans;

ans = getGcdMulti(n, value);

printf("GCD= %d\n", ans);

}

int getGCD(int x, int y)

{

while(x%y!=0)

{

int g=x%y;

x=y;

y=g;

}

return y;

}

int getGcdMulti(int n, int value[])

{

int gcd= value[0];

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

{

gcd= getGCD(value[i], gcd);

}

return gcd;

}

**Q-4.** void makeSum(int a, int b, int \*sum)

The above function takes as input two integers a and b and stores their sum in the variable sum. Implement the function.

**Ans:**

#include <stdio.h>

void makeSum(int a, int b, int \*sum);

void main()

{

int a, b;

scanf("%d %d", &a, &b);

int sum;

makeSum(a, b, &sum);

printf("Sum= %d\n", sum);

}

void makeSum(int a, int b, int \*sum)

{

\*sum= a+b;

}

**Q-5.** Find the output of the following code:

int a = 10;

int b = 20;

int \*p = &a;

printf(“%d %d %d\n”, a, b, \*p);

\*p = 30;

printf(“%d %d %d\n”, a, b, \*p);

\*p = 20;

a = 50;

b = 10;

printf(“%d %d %d\n”, a, b, \*p);

p = &b;

printf(“%d %d %d\n”, a, b, \*p);

a = 20;

b = 100;

printf(“%d %d %d\n”, a, b, \*p);

**Ans:**

**10 20 10**

**30 20 30**

**50 10 50**

**50 10 10**

**20 100 100**

Q-6. void makeNArray(int n, int squares[]) populates squares with first n square numbers. Implement the function.

(For example, makeNarray(4, squares) will populate squares with 1, 4, 9, 16.)

**Ans:**

#include <stdio.h>

void makeNArray(int n, int squares[]);

void main()

{

int n, squares[n];

scanf("%d", &n);

makeNArray(n, squares);

}

void makeNArray(int n, int squares[])

{

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

{

squares[i]=(i+1)\*(i+1);

}

printf("The square of first %d natural number is ", n);

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

{

printf("%4d", squares[i]);

}

}

**Q-7.** How can you use the return by address mechanism to return two integers from a function? Explain with an example.

## Q-8. stringCat(char a[], char b[], char out[]) The strcat function takes two strings and concats them and outputs them to a out. Assume that out has enough space to hold contents of both a and b. Implement the function.

**Ans:**

#include <stdio.h>

#include <string.h>

void stringCat(char a[], char b[], char out[]);

void main()

{

char a[100], b[100], out[200];

gets(a);

gets(b);

stringCat(a, b, out);

}

void stringCat(char a[], char b[], char out[])

{

out=strcat(a,b);

printf("%s", out);

}

**Q-9.** Write a program using pointers to read in an array of integers and Print its elements in reverse order.

**Ans:**

#include <stdio.h>

void RevArray(int \*n, int array[]);

int main()

{

int n;

scanf("%d", &n);

int a[n];

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

{

scanf("%d", &a[i]);

}

RevArray(&n, a);

return 0;

}

void RevArray(int \*n, int array[])

{

for(int i=0, j=\*n-1; i<j; i++, j--)

{

int temp= array[i];

array[i]= array[j];

array[j]= temp;

}

for(int i=0; i<\*n; i++)

{

printf("%d ", array[i]);

}

}

**Q-10.** The main is a user-defined function. How does it differ from other user-defined functions?

The main () function differ from other user-defined functions in the following ways:

1. The compilers of most of the programming languages are so designed that the main () function constitutes the entry point of the program execution. It defines the point from which the program has to start executing itself though there are many other sub-routines and other user-defined functions included in the program.
2. The main () function is the controlling section of our code because even though the control of the program is shifted to the UDF (user defined function) during the program execution after a function-call from main (), once it's execution is completed, the control is transferred back to the main () function with some or no return value (as in the case of a void function).
3. The main () function provides a platform for calling the first user-defined function in the program.
4. It has got its own functionality and structural features with respect to the usage of syntaxes which cannot be changed by the end user unless he writes his own compiler. But the UDF's have functions and structures designed by the user or programmer.

**How can a function be declared that takes an integer and a double as input and returns a char in c programming**