1. Write a function that takes as input a number and outputs its square.
2. How do you declare a function that takes as input an integer and a double and returns a char?
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. You can assume that the gcd function is implemented.
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.

1. 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);

1. void makeNArray(int n, int squares[]) populates squares with first n square numbers. For example, makeNarray(4, squares) will populate squares with 1, 4, 9, 16. Implement the function.
2. How can you use the return by address mechanism to return two integers from a function? Explain with an example.
3. stringCat(char a[], char b[], char out[]) The strcat function takes two strings and contacts them and outputs them to a out. Assume that out has enough space to hold contents of both a and b. Implement the function.
4. Write a program using pointers to read in an array of integers and Print its elements in reverse order.
5. The main is a user-define function. How does it differ from other user-defined functions?

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

**Ans:** we know in the c language we cannot return multiple values from a function. But fortunately, we have pointers by which we can solve this problem. Using the ‘return by address’ mechanism of pointers, we can return two integers from a function.

Below is a program to swap two integer values of two variables using a pointer. By passing the address of the two variables as function arguments we can swap values and make changes in the actual variables.

//code

#include<stdio.h>

#include<string.h>

// function definition

void swap(int\* a, int\* b){

int temp=\*a;

\*a=\*b;

\*b=temp;

}

// driver code

int main(){

int a=10,b=20;

swap(&a,&b);

printf("%d %d",a,b);

return 0;

}

Here, swap is a user-defined function, in the main function “&a” and “&b” are passing addresses of “a” and “b” variables to the swap function. In the swap function, there are two integer type parameters \* a and \*b, which denotes that “a” and “b” are pointer-type variables, also \* a and \*b holds the values of the actual variables “a” and “b”. \* a and \*b values are swapped and returned to the main function.

Thus by using the return by address mechanism of pointers, we can return two integers from a function.

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

**Ans:** although main() is a user-defined function, it has some characteristics which have made it different from other user-defined functions.

* 1. The main() function is the entry point of the c compiler. Code starts executing from here and then other functions are called according to the requirements.
  2. This main() is the controlling section of the code execution, although after calling a user-defined function control is shifted to the main(), and after executing the user-defined function, control is transferred back to main() function.
  3. If there are any user-defined functions in the code, main() function is the platform from where the first user-defined function can be called.
  4. In the c99 and later, if a program comes at the end of the main() function without an explicit return statement, it is equivalent to return 0. No other function has this characteristics.