**Fundamentals of Computer Programming**

**ASSIGNMENT**

**IV**

**Prepared by: Rupesh Pandit**

**Shift: Morning**

**Roll Number: 18**

**BSc CSIT**

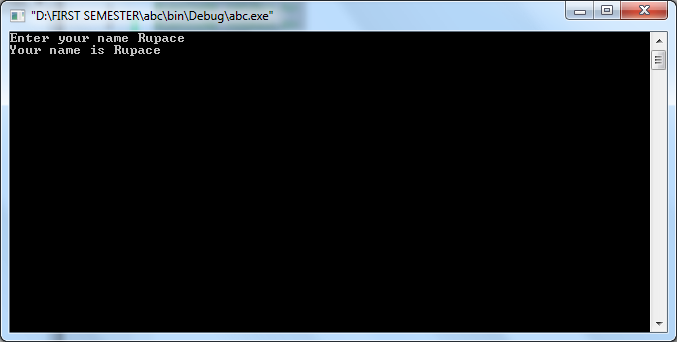
**Character array and String**

1. **WAP to input and output of String using Scanf() and Printf()**

**Code:**

|  |
| --- |
| **/\***  **PROGRAM: To input and output of String using Scanf() and Printf()**  **\*/**  **#include <stdio.h>**  **#include <conio.h>**  **void main()**  **{**  **char YourName[10];**  **printf("Enter your name ");**  **scanf("%s",YourName);**  **printf("Your name is %s",YourName);**  **getch();**  **}** |

**Output:**

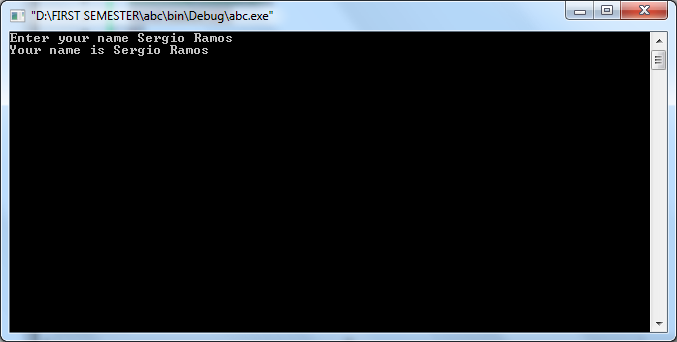


1. **WAP to input and output of String using gets() and Puts().**

**Code:**

|  |
| --- |
| **/\***  **PROGRAM: To input and output of String using gets() and Puts().**  **\*/**  **#include<stdio.h>**  **#include<conio.h>**  **void main ()**  **{**  **char Name[10];**  **printf("Enter your name ");**  **gets(Name);**  **printf("Your name is ");**  **puts(Name);**  **getch();**  **}** |

**Output:**

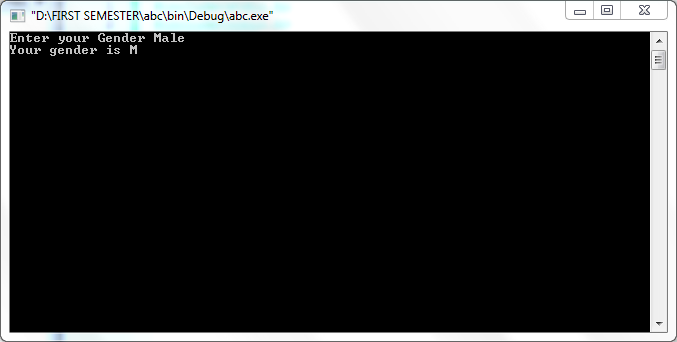


1. **WAP to input and output of Character array using getchar() and Putchar().**

**Code:**

|  |
| --- |
| **/\***  **PROGRAM: To input and output of character using getchar() and Putchar().**  **\*/**  **#include<stdio.h>**  **#include<conio.h>**  **void main ()**  **{**  **char Gender;**  **printf("Enter your Gender ");**  **Gender = getchar();**  **printf("Your gender is ");**  **putchar(Gender);**  **getch();**  **}** |

**Output:**



1. **WAP to input and string and find out the length of string using strlen() and without using strlen().**

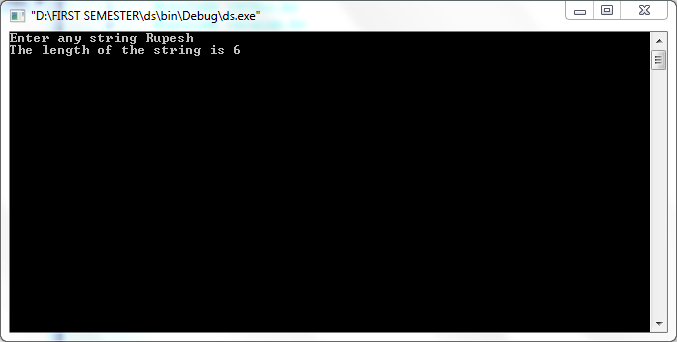
**Code:**

**With strlen():**

|  |
| --- |
| **/\***  **PROGRAM: input and string and find out the length of string using strlen() \*/**  **#include <stdio.h>**  **#include <stdlib.h>**  **void main()**  **{**  **char String[50];**  **printf("Enter any string ");**  **scanf("%s",&String);**  **printf("The length of the string is %d",strlen(String));**  **getch();**  **}** |

|  |
| --- |
| **/\*Without strlen()**  **PROGRAM: To input and string and find out the length of string without using strlen() \*/**  **#include<stdio.h>**  **#include<conio.h>** |

**Output:**



1. **WAP to copy a string from one to another using strcpy() and without using strcpy().**

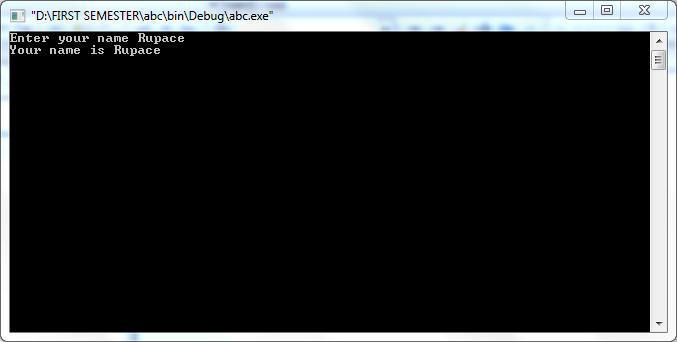
**Code:**

**With strcpy():**

|  |
| --- |
| **/\***   1. **PROGRAM: To copy a string from one to another using strcpy().**   **\*/**  **#include<stdio.h>**  **#include<conio.h>**  **void main ()**  **{**  **char Name1[10],Name2[10];**  **printf("Enter your name ");**  **gets(Name1);**  **strcpy(Name2, Name1);**  **printf("Your name is ");**  **puts(Name2);**  **getch();**  **}** |

|  |
| --- |
| **/\***   1. **PROGRAM: To copy a string from one to another without using strcpy().**   **\*/**  **#include<stdio.h>**  **#include<conio.h>** |

**Output:**



1. **WAP to concatenate two string using strcat() and without using strcat()**

**Code:**

**With strlen():**

|  |
| --- |
| **/\***  **PROGRAM: To concatenate two string using strcat() \*/**  **#include <stdio.h>**  **#include <stdlib.h>** |

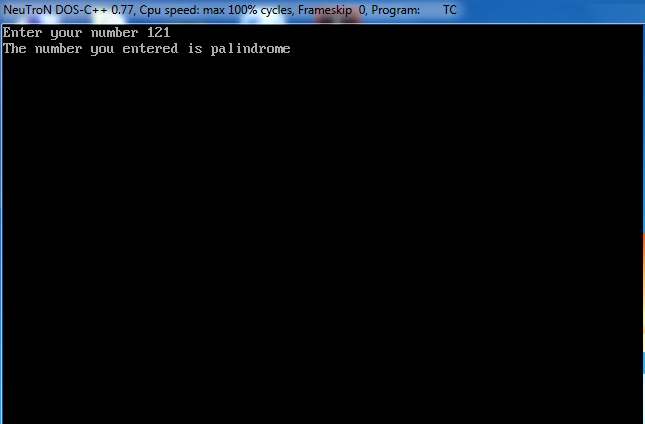
|  |
| --- |
| **/\*Without strcat()**  **PROGRAM: To to concatenate two string without using strcat()\*/**  **#include<stdio.h>**  **#include<conio.h>** |

1. **WAP to compare two string using strcmp() and without using Strcmp().**
2. **WAP to reverse a string using strrev() and without using strrev().**
3. **WAP to find whether the given string is palindrome or not.**

**Code:**

|  |
| --- |
| **/\***  **PROGRAM: To find whether the given string is palindrome or not. ()**  **\*/**  **#include<stdio.h>**  **#include<conio.h>**  **void main()**  **{**  **int YourNumber,rev=0,temp=0,Rem=0;**  **clrscr();**  **printf("Enter your number ");**  **scanf("%d",&YourNumber);**  **temp=YourNumber;**  **while(YourNumber!=0){**  **Rem=YourNumber%10;**  **rev = (rev\*10) + Rem;**  **YourNumber = YourNumber/10;**  **}**  **if(rev==temp){**  **printf("The number you entered is palindrome");**  **}**  **else**  **printf("The number you entered is not palindrome");**  **getch();**  **}** |

**Output:**



1. **WAP to input the name of any five student and sort them alphabetically in ascending order.**

**Code:**

|  |
| --- |
| **/\***   1. **PROGRAM: To input the name of any five student and sort them alphabetically in ascending order.**   **\*/** |

WAP to input a string and convert it into upper case and vice versa.

WAP to take the string using gets(), and pass to the function to find and return number of words, whitespace in that string.

Write a program to read a string and search a specified word in given string.

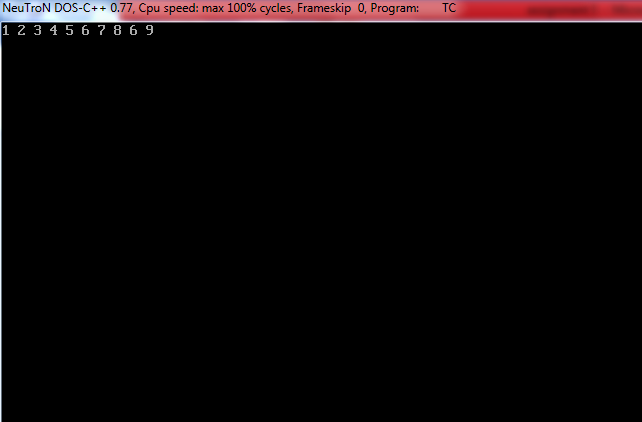
### Arrays and Strings:

1. WAP in C to initialize any 10 value in an array and display them.

**Code:**

|  |
| --- |
| **/\***  **PROGRAM: To** initialize any 10 value in an array and display them **\*/**  **#include <stdio.h>**  **#include <stdlib.h>**  **void main()**  **{**  **int nums[10]={1,2,3,4,5,6,7,8,6,9},i;**  **for(i=0;i<10;i++){**  **printf("The numbers are %d ",nums[i]);**  **}**  **getch();**  **}** |

Output:

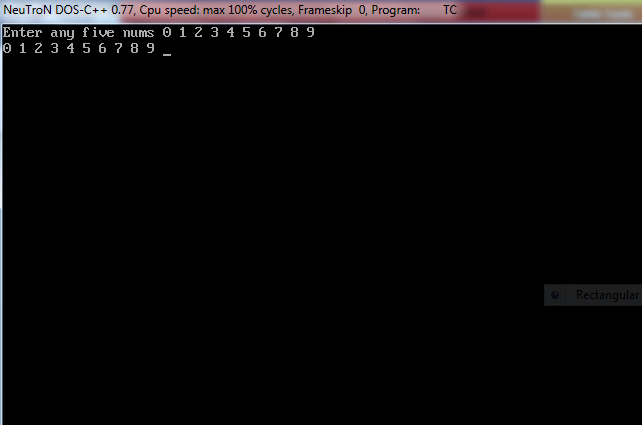


1. WAP to input any 10 number by user and display them.

**Code:**

|  |
| --- |
| **/\***  **PROGRAM: To** input any 10 number by user and display them.**\*/**  **#include <stdio.h>**  **#include <stdlib.h>**  **void main()**  **{**  **int nums[10],i;**  **printf("Enter any five nums ");**  **for(i=0;i<10;i++){**  **scanf("%d ",&nums[i]);**  **}**  **for(i=0;i<10;i++){**  **printf("The numbers are %d ",nums[i]);**  **}**  **getch();**  **}** |

Output:

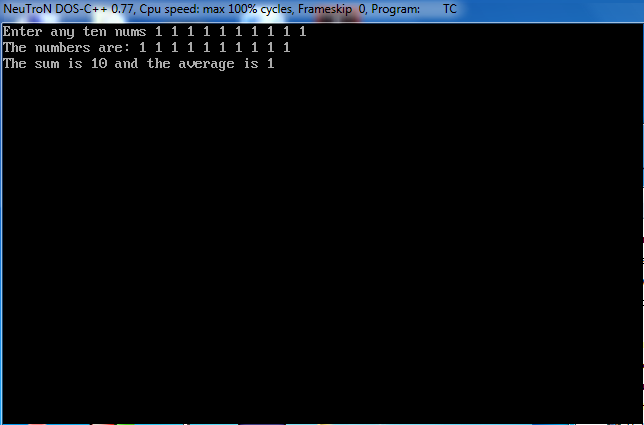


1. WAP in C to input any 10 number in an array and print them along with the Total and average of that numbers.

**Code:**

|  |
| --- |
| **/\***  **PROGRAM: To** to input any 10 number in an array and print them along with the Total and average of that numbers.  **\*/**  **#include <stdio.h>**  **#include <stdlib.h>**  **void main()**  **{**  **int nums[10],i,sum=0,avg=0;**  **clrscr();**  **printf("Enter any ten nums ");**  **for(i=0;i<10;i++){**  **scanf("%d",&nums[i]);**  **sum=sum+nums[i];**  **avg=sum/10;**  **}**  **printf("The numbers are: ");**  **for(i=0;i<10;i++){**  **printf("%d ",nums[i]);**  **}**  **printf("\nThe sum is %d and the average is %d",sum,avg);**  **getch();**  **}** |

Output:

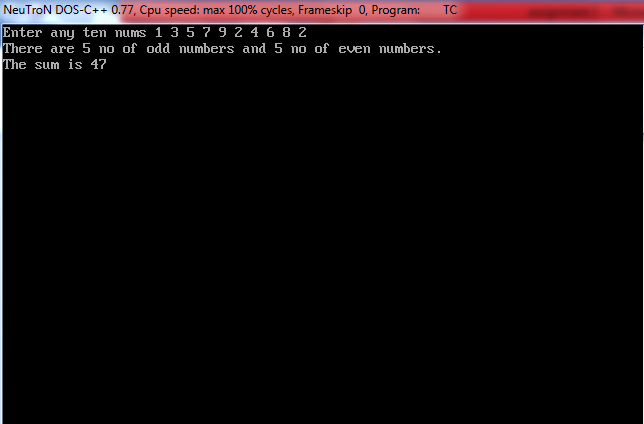


1. WAP in C to input any 10 numbers in an array and count no. of ODD and EVEN and find out their sum and display them.

**Code:**

|  |
| --- |
| **/\***  **PROGRAM: To** to input any input any 10 numbers in an array and count no. of ODD and EVEN and find out their sum and display them.  **\*/**  **#include <stdio.h>**  **#include <stdlib.h>**  **void main()**  **{**  **int nums[10],i,sum=0,even=0,odd=0;**  **clrscr();**  **printf("Enter any ten nums ");**  **for(i=0;i<10;i++){**  **scanf("%d",&nums[i]);**  **sum=sum+nums[i];**  **if(nums[i]%2==0){**  **odd=odd++ ;**  **if(nums[i]!=0){**  **even=even++;**  **} }**  **}**  **printf("There are %d no of odd numbers and %d no of even numbers.",odd,even);**  **printf("\nThe sum is %d",sum);**  **getch();**  **}** |

Output:

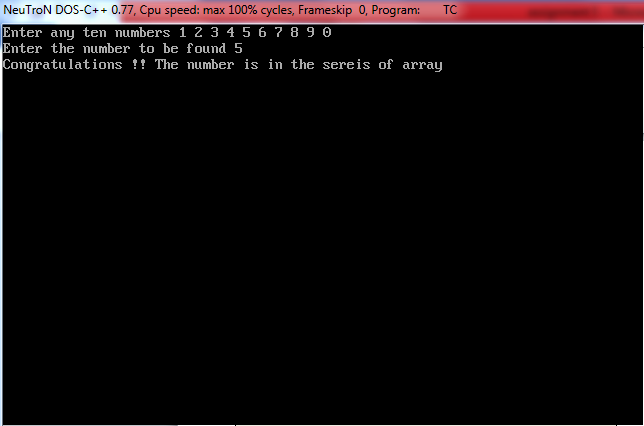


1. WAP in C to input any 10 numbers in an array and search an element.

**Code:**

|  |
| --- |
| **/\***  **PROGRAM: To** to input any 10 numbers in an array and search an element **\*/**  **#include<stdio.h>**  **#include<conio.h>**  **void main()**  **{**  **int i,n,nums[10],temp;**  **clrscr();**  **printf("Enter any ten numbers ");**  **for(i=0;i<10;i++){**  **scanf("%d",&nums[i]);**  **}**  **printf("Enter the number to be found ");**  **scanf("%d",&n);**  **for(i=0;i<10;i++){**  **if(n==nums[i]){**  **temp=temp++;**  **}**  **}**  **if(temp!=0){**  **printf("Congratulations !! The number is in the sereis of array ");}**  **else{**  **printf("The number could not be found") ;**  **}**  **getch();**  **}** |

Output::

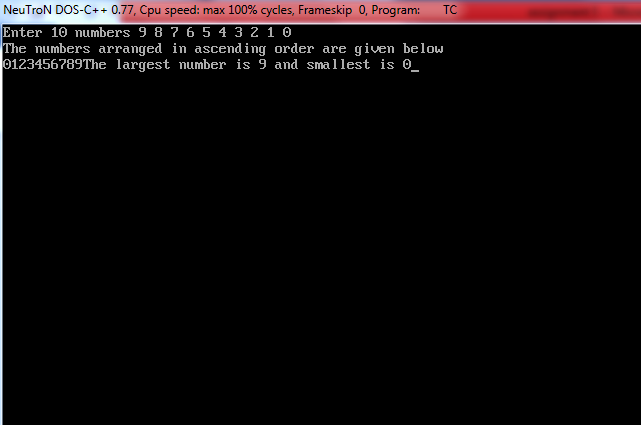


1. WAP to input any 10 numbers in an array and find out the Maximum and Minimum Value.

**Code:**

|  |
| --- |
| **/\***  **PROGRAM: To** to input any 10 numbers in an array and find out the Maximum and Minimum Value.  **\*/**  **#include<stdio.h>**  **#include<conio.h>**  **void main()**  **{**  **int i,j,temp=0,number[10];**  **clrscr();**  **printf("Enter 10 numbers ");**  **for(i=0;i<10;i++)**  **scanf("%d",&number[i]);**  **for(i=0;i<9;i++)**  **{**  **for(j=i+1;j<10;j++)**  **{**  **if(number[i]>number[j])**  **{**  **temp=number[i];**  **number[i] = number[j];**  **number[j] = temp;**  **}**  **}**  **}**  **printf("The numbers arranged in ascending order are given below \n");**  **for (i = 0; i <=9; i++){**  **printf("%d",number[i]);**  **}**  **printf("The largest number is %d and smallest is %d",number[9],number[0]);**  **getch();**  **}** |

Output:

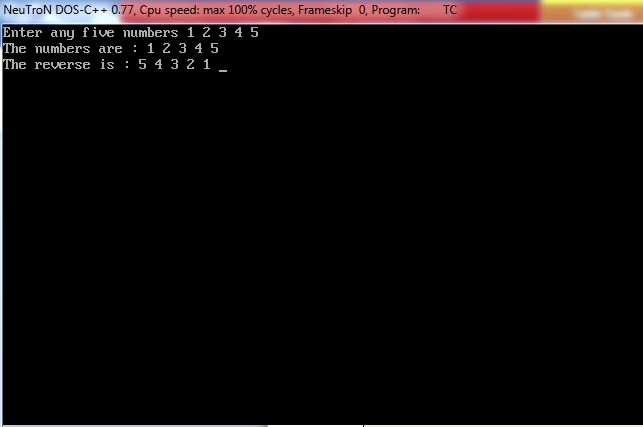


1. WAP in C to convert decimal number to Binary, Octal, and Hexadecimal using array.
2. WAP to input any 5 elements in an array and print them in reverse order too.

**Code:**

|  |
| --- |
| **/\***  **PROGRAM: To** input any 5 elements in an array and print them in reverse order too.  **\*/**  **#include<stdio.h>**  **#include<conio.h>**  **void main()**  **{**  **int i,n,nums[5];**  **clrscr();**  **printf("Enter any five numbers ");**  **for(i=0;i<5;i++){**  **scanf("%d",&nums[i]);**  **}**  **printf("The numbers are : ");**  **for(i=0;i<5;i++){**  **printf("%d ",nums[i]);**  **}**  **printf("\nThe reverse is : ");**  **for(i=4;i>=0;i--){**  **printf("%d ",nums[i]);**  **}**  **getch();**  **}** |

Output:



**THEORITICAL ASSIGNMENTS:**

1. Explain in detail about storage class. Write in detail about 4 different storage class.

A storage class defines the scope visibility and life-time of variables and/or functions within a C Program. These specifiers precede the type that they modify. There are the following storage classes, which can be used in a C Program auto register static extern:

* Auto
* Register
* Static
* Extern

Auto Storage class: The auto storage class is the default storage class for all local variables.

Example: { int mount;

auto int month; }

The example above defines two variables with the same storage class, auto can only be used within functions, i.e., local variables.

Register Storage class: The register storage class is used to define local variables that should be stored in a register instead of RAM. This means that the variable has a maximum size equal to the register size and can't have the unary '&' operator applied to it .

Example: { register int miles; }

Static Storage class: The static storage class instructs the compiler to keep a local variable in existence during the lifetime of the program instead of creating and destroying it each time it comes into and goes out of scope. Therefore, making local variables static allows them to maintain their values between function calls. The static modifier may also be applied to global variables. When this is done, it causes that variable's scope to be restricted to the file in which it is declared. In C programming, when static is used on a class data member, it causes only one copy of that member to be shared by all objects of its class.

Extern Storage class: The extern storage class is used to give a reference of a global variable that is visible to ALL the program files. When you use 'extern', the variable cannot be initialized as all it does is point the variable name at a storage location that has been previously defined. When you have multiple files and you define a global variable or function, which will be used in other files also, then extern will be used in another file to give reference of defined variable or function. Just for understanding, extern is used to declare a global variable or function in another file.

1. Write one simple example of for-loop. Explain about the steps how it works.

Working Principle :

The initialization statement is executed only once.

Then, the test expression is evaluated. If the test expression is false (0), for loop is terminated. But if the test expression is true (nonzero), codes inside the body of for loop is executed and the update expression is updated.

This process repeats until the test expression is false.

The for loop is commonly used when the number of iterations is known.

To learn more on test expression (when test expression is evaluated to nonzero (true) and 0 (false)), check out [relational](https://www.programiz.com/c-programming/c-operators#relational) and [logical operators](https://www.programiz.com/c-programming/c-operators#logical).

#include<stdio.h>

#include<conio.h>

int main()

{

int num, count, sum = 0;

printf("Enter a positive integer: ");

scanf("%d", &num);

for(count = 1; count <= num; ++count)

{

sum += count;

}

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

return 0;

}

1. Write one simple example of while-loop. Explain about the steps how it works.

Working Principle:

The while loop evaluates the test expression.If the test expression is true (nonzero), codes inside the body of while loop is evaluated. The test expression is evaluated again. The process goes on until the test expression is false.When the test expression is false, the while loop is terminated.

#include<stdio.h>

#include<conio.h>

int main()

{

int number;

long long factorial;

printf("Enter an integer: ");

scanf("%d",&number);

factorial = 1;

while (number > 0)

{

factorial \*= number;

--number;

}

printf("Factorial= %lld", factorial);

return 0;

}

1. Write a paragraph and an example of each about strlen(), strcpy(), strcat(), strcmp(), strrev().

**strlen**():

The C library function size\_t strlen(const char \*str) computes the length of the string str up to, but not including the terminating null character.

Syantax: size\_t strlen(const char \*str)

[ str -- This is the string whose length is to be found.]

**strcpy():**

The C library function char \*strcpy(char \*dest, const char \*src) copies the string pointed to, by src to dest.

Syntax: char \*strcpy(char \*dest, const char \*src)

[ dest -- This is the pointer to the destination array where the content is to be copied. src -- This is the string to be copied.]

**strcat() :**

The C library function char \*strcat(char \*dest, const char \*src) appends the string pointed to by src to the end of the string pointed to by dest.

Syntax: char \*strcat(char \*dest, const char \*src)

[dest -- This is pointer to the destination array, which should contain a C string, and should be large enough to contain the concatenated resulting string. src -- This is the string to be appended. This should not overlap the destination.]

**strcmp():**

The C library function int strcmp(const char \*str1, const char \*str2) compares the string pointed to, by str1 to the string pointed to by str2.

Syntax: int strcmp(const char \*str1, const char \*str2)

[str1 -- This is the first string to be compared.str2 -- This is the second string to be compared.]

**strrev():**

strrev( ) function reverses a given string in C language. Syntax for strrev( ) function is given below.  
char \*strrev(char \*string); strrev( ) function is non standard function which may not available in standard library in C.

1. What is array? How array is different from ordinary variable.

Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables. A specific element in an array is accessed by an index.

All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.

type arrayName [ arraySize ];

DIFFERENCE BETWEEN ARRAY AND ORDINARY VARIABLE

* Array is the set of an multiple values where as variable can store single value at a time.
* Array are defined using subscript where as ordinary variable do not need any subscripting.
* Values are always extracted through looping whereas we take the inputs step by step in ordinary variable.

1. How can you prepare a student mark sheets software using the things you’ve learned till now? Give me the list of the things you’ve learned and how can you use it to do what kind of operation to prepare a mark sheet of a student.

We can prepare a students marksheets through c language using loops nad other arrays and operators. Different loops and operators are required to make a students marksheet for example for loop, +, -,= ,etc are used.

Firstly , the name of students are extracted form the user for the further calculations. The names of each and every student is lengthy process so for that the name is scanned through loop of desired no of students. After we get their name others details like class roll number are also gained through the similar process. After that another loop is crated in which we get the inputs from the user about the marks obtained by them in different subjects. The total is calculated from assigning the values of sum of the score in each and every subject they have entered. Percentage is also calculated by dividing the sum by the number of the subjects.

[REQUIRED ]: Respective variables and their data types, for loop, printf, scanf.