Pere-defined function in c:-

(1) strlen():

The strlen() function in c is used to calculate the length of a string by counting the number of characters before the terminating null character (\0). It is part of the <string.h> library.

(2) strcpy():

The strcpy() function in c is used to copy the contents of a source string to a destination string. It is defined in the <string.h> header file and copies characters, including the null-terminator (\0), until the end of the source string is reached.

(3) The strncpy():

The strncpy() function in c copies a specified number of characters from a source string to a destination buffer. It is defined in the <string.h> header file.

(4) strncat():

The strncat function in c is used to concatenate (append) a specific number of characters from a source string to the end of a destination string. It is defined in the <string.h> header file and is often preferred over strcat() because it allows you to limit the number of characters appended, which helps prevent buffer overflow.

(5) strcat():

The strcat() function in c is used to concatenate (join) two strings. It appends the source string to the end of destination string, modifying the destination string in place.

(6) strcmp():

The strcmp() function in c is used to compare two null-terminated strings lexicographically (alphabetical order, based on ASCII values). It is defined in the <string.h> header file and returns an integer value to indicate the relationship between the string.

strncmp():

The strncmp() function in c is used to compare a specific number of characters (a prefix) of two strings. The comparision is lexicographical, based on the ASCII values of the characters.

(1) Code for number of vowels, consonents and digits.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
  char str[200];
    int i, vowels=0, consonants=0 ; digits=0;
  Printf("Enter a string:");
  fgets (str,sizeof(str),stdin);
  for(i=0; str[i]!='10';i++)
{ charch=tolower(str[i]);
  if (ch>='0' && ch<='9'){
  digits++;
}
  else if (ch>='a' ff ch<='z')
{if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u')
  vowels++;
  else
    cosonants++;
}
  }
    printf ("vowels: %d\n", vowels);
    printf ("consonants : %d\n", consonants);
    printf ("Digits: %d\n", digits);
  return0;
}
```

Output:

```
Enter a string : Hello world 123
    vowels: 3
    consonants: 7
    Digits: 3
```