Ex. No19

Store a Character String using Memory Management Functions

Aim:

To write a C program to allocate and reallocate memory block for character string.

Algorithm:

1) Start the program.
2) Declare two variables name[100] and *desc.
3) Copy a name to the variable “name”.
4) Using “malloc” function allocate memory to “desc” variable.
5) Check if “desc” equal to “NULL”
   a). Display “unable to allocate memory”.
6) Else
   a). Copy description text to “desc” variable.
7) Using “realloc” function allocate higher memory to “desc” variable.
8) Check if “desc” equal to “NULL”
   a). Display “unable to allocate memory”.
9) Else
   a). Concat additional description text to “desc” variable.
10) Display Name and Description.
11) Using “free” function free the memory allocated to “desc” variable.
12) Stop the program.

Program:

```
#include <stdio.h>
#include<conio.h>
#include <alloc.h>
#include <string.h>

void main()
{
  char name[100], *desc;
  clrscr();
  strcpy(name, "Rajesh");
  desc = malloc( 40 * sizeof(char) );
  if( desc == NULL )
  {
   printf("Error - unable to allocate required memory\n");
  }
  else
  {
   strcpy( desc, "Rajeh is studying Computer Science.");
  }
  desc = realloc( desc, 100 * sizeof(char) );
  if( desc == NULL )
  {
   printf("Error - unable to allocate required memory\n");
  }
  else
  {
   strcat( desc, " In Tamil Nadu Polytechnic College, Madurai.");
  }
  printf("Name = %s\n", name );
  printf("Description: %s\n", desc );
  free(desc);
  getch();
}
```

Result:

Thus the C program to allocate and reallocate memory block for character string is executed successfully.