```c
#include <stdio.h>
#include <stdlib.h>

#define MAX_STRING_LENGTH 512

int main(void)
{
    //function declarations
    char* ReplaceVowelsWithHashSymbol(char *);

    //variable declaration
    char string[MAX_STRING_LENGTH];
    char *replaced_string = NULL;

    //code
    printf("\n\n");
    printf("Enter String : ");
    gets_s(string, MAX_STRING_LENGTH);

    replaced_string = ReplaceVowelsWithHashSymbol(string);
    if (replaced_string == NULL)
    {
        printf("ReplaceVowelsWithHashSymbol() Function Has Failed !!! Exitiing Now...\n\n");
        exit(0);
    }

    printf("\n\n");
    printf("Replaced String Is : \n\n");
    printf("%s\n\n", replaced_string);

    if (replaced_string)
    {
        free(replaced_string);
        replaced_string = NULL;
    }

    return(0);
}

char* ReplaceVowelsWithHashSymbol(char *s)
{
    //function prototype
    void MyStrcpy(char *, char *);
    int MyStrlen(char *);

    //varibale declarations
    char *new_string = NULL;
    int i;

    //code
    new_string = (char *)malloc(MyStrlen(s) * sizeof(char));
    if (new_string == NULL)
    {
        printf("COULD NOT ALLOCATE MEMORY FOR NEW STRING !!!\n\n");
        return(NULL);
```

```
56      }
57
58      MyStrcpy(new_string, s);
59      for (i = 0; i < MyStrlen(new_string); i++)
60      {
61          switch (new_string[i])
62          {
63          case 'A':
64          case 'a':
65          case 'E':
66          case 'e':
67          case 'I':
68          case 'i':
69          case 'O':
70          case 'o':
71          case 'U':
72          case 'u':
73              new_string[i] = '#';
74              break;
75          default:
76              break;
77          }
78      }
79
80      return(new_string);
81  }
82
83  void MyStrcpy(char *str_destination, char *str_source)
84  {
85      //function prototype
86      int MyStrlen(char *);
87
88      //variable declarations
89      int iStringLength = 0;
90      int j;
91
92      //code
93      iStringLength = MyStrlen(str_source);
94      for (j = 0; j < iStringLength; j++)
95          *(str_destination + j) = *(str_source + j);
96
97      *(str_destination + j) = '\0';
98  }
99
100 int MyStrlen(char *str)
101 {
102     //variable declarations
103     int j;
104     int string_length = 0;
105
106     //code
107     // *** DETERMINING EXACT LENGTH OF THE STRING, BY DETECTING THE FIRST
        OCCURENCE OF NULL-TERMINATING CHARACTER ( \0 ) ***
108     for (j = 0; j < MAX_STRING_LENGTH; j++)
109     {
110         if (str[j] == '\0')
```

```
111             break;
112         else
113             string_length++;
114     }
115     return(string_length);
116 }
117
118
```