PPS CODES

NAME:K.Santhanalakshmi

REG NO:-RA2011027010129

1. Identify the issue in the following program

```
void main() {
int *p = (int *) malloc(sizeof(int)); p=NULL;
free(p); }
```

free() can be called for NULL pointer, so no problem with free function call.

The problem is memory leak, p is allocated some memory which is not freed, but the pointer is assigned as NULL. The correct sequence should be following:

```
free(p);
 p = NULL;
```

---

2. Explain function call to other function with simple example.

We can can call a function into another function for that we just need to pass necessary argument and correct return values

Example:-

```
#include <stdio.h>
#include <stdlib.h>
int add_two_ints(int x,int y);

int main(void)
{
    int a,b,c;
    scanf("%d %d",&a,&b);
    //calling the function
```

```
        c = add_two_ints(a,b);
        printf("sum of %d and %d is equal to %d",a,b,c);
}

int add_two_ints(int x,int y)
{
        return x+y;
}
```

In above example we declare a function first and as usual we declare main function and scanf statements and after that we call the function-int add_two_ints(a,b) into the main function, after printing statements we pass the function- int_add_two_ints(a,b) out side the main function and we will return the value like return (x+y); to the main function. Compiler takes it like user has made a copy of the function before main function and he is calling that function into main function whose operating statement is going to be after main function, so output addition operation will be printed.

3.Write a C program to sort array of strings in non decreasing order of their length. If two strings have the same length, then the lexicographically smaller string should appear first. Function prototype should be int sortby_length(char *, char *)

Sample input:

4

xrt

asg

ghtyu

hjuk

Output

asg

xrt

hjuk

ghtyu

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int sort_by_length(const char* a, const char* b){
   if(strlen(a) != strlen(b))
      return strlen(a) > strlen(b);
   else
      return strcmp(a, b) >  0;
}

void string_sort(char** arr,const int len,int (*cmp_func)(const char*
a, const char* b)){
   int i;
       for( i = 1; i < len; i++){
      int j = i;
      char* p = arr[i];

      while(j > 0){
         if((*cmp_func)(arr[j-1],p) > 0 )
            arr[j] = arr[j-1];
         else
            break;
```

```
            j--;
        }
        arr[j] = p;

    }
}


int main()
{
    int n;
    scanf("%d", &n);



    char** arr;
         arr = (char**)malloc(n * sizeof(char*));
  int i;
    for( i = 0; i < n; i++){
        *(arr + i) = malloc(1024 * sizeof(char));
        scanf("%s", *(arr + i));
        *(arr + i) = realloc(*(arr + i), strlen(*(arr + i)) + 1);
    }
  printf("\n");
```

```
    string_sort(arr, n, sort_by_length);
         for( i = 0; i < n; i++)
        printf("%s\n", arr[i]);
    printf("\n");
}
```
