Name: Jasmine Jones

Question 1: Write two Simple Programs in C that uses the Exit Library function. In each program print some outputs and explain your results.

1st program:

```
#include <stdio.h>
#include <stdlib.h>
int main()
        int i;
        for(i = 0; i <= 7; i++){
                if(i == 5){
                        exit(0);
}
else{
        printf("%d\n" , i);
        }
}
return 0;
}
```

2nd program:

```
#include <stdio.h>
#include <stdlib.h>
int main(){
        int num;
        print f("Please enter a number that is positive");
        scanf("%d", num);
```

```
if(num < 0){
        printf("Error: you have just entered a negative number);
        exit(1);
}
printf("Great Job, you have entered %d. \n", num);
exit(0);
}
```

Question 2: Modify program in listing 11:4 to handle four students in your group. Your program should take in the firstname and lastname of the two students, their phone numbers. Print the results

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

struct student {
   char firstName[30];
   char lastName[30];
   char phone[15];
};

int main() {
   int i, n = 4;
```

```
struct student *students;


students = (struct student*) malloc(n * sizeof(struct student));
if (!students) {
   printf("failed!\n");
   return 1;
}


for (i = 0; i < n; i++) {
   printf("Please enter first name of student %d: ", i + 1);
   scanf("%s", students[i].firstName);


   printf("Enter last name of student %d: ", i + 1);
   scanf("%s", students[i].lastName);


   printf("Enter phone number of student %d: ", i + 1);
   scanf("%s", students[i].phone);
   printf("\n");
}


printf("--- Student Information ---\n");
for (i = 0; i < n; i++) {
```

```
        printf("Student %d: %s %s, Phone: %s\n", i+1, students[i].firstName, students[i].lastName,
students[i].phone);
    }

    free(students);
    return 0;
}
```

Question 3: Modify your program in (2) to demonstrate stepping through an array of structures using pointer notations *pointer like in listing 11.5

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

// Define a structure for student info
struct student {
    char firstName[30];
    char lastName[30];
    char phone[15];
};

int main() {
    int i, n = 4; // 4 students
```

```
struct student *students, *ptr;

// Allocate memory for n students
students = (struct student*) malloc(n * sizeof(struct student));
if (students == NULL) {
    printf("Memory allocation failed!\n");
    return 1;
}

// Use a pointer to step through the array
ptr = students;

// Input student details using pointer notation
for (i = 0; i < n; i++) {
    printf("Enter first name of student %d: ", i + 1);
    scanf("%s", ptr->firstName);

    printf("Enter last name of student %d: ", i + 1);
    scanf("%s", ptr->lastName);

    printf("Enter phone number of student %d: ", i + 1);
    scanf("%s", ptr->phone);
```

```
        printf("\n");
        ptr++; // move pointer to next student
    }

    // Reset pointer to start of array to print details
    ptr = students;
    printf("--- Student Information ---\n");
    for (i = 0; i < n; i++) {
        printf("Student %d: %s %s, Phone: %s\n",
               i + 1,
               ptr->firstName,
               ptr->lastName,
               ptr->phone);
        ptr++; // move to next student
    }

    // Free dynamically allocated memory
    free(students);

    return 0;
}
```

Question 4: What is puts and gets in C standard library. Give a simple example of how to use it.

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

```c
#include <stdio.h>

int main() {
    char name[50];

    puts("Enter your name:");
    fgets(name, sizeof(name), stdin); // safer than gets()

    puts("You entered:");
    puts(name);

    return 0;
}
```