Week - 1 Programming Assignments

Arrays, Strings & Pointers

To submit this assignment, all students should group all C-script files in '.zip' format with the zip file formatted as:

```
STUDENT_ID_ASSIGNMENT_ID.zip
```

ALL C-script files should also be formatted as:

```
ASSIGNMENT_ID_QUESTION_ID_STUDENT_ID.c
```

To submit this assignment, all students should group all c-script files in '.zip' format with the zip file formatted as:

Example 1: Array

Given an array 'arr', determine the smallest number in the array

Constraints:

- The first input is the length of the array
- The following inputs are the numbers in the array

Example 1:

|Input: 3 1 2 3
|Output: 1

Example 2:

|Input: 5 4 3 6 2 5
|Output: 2

```c
#include <stdio.h>
//Write a program to determine the smallest number in the array
int smallest(int arr[], int n){
    int i;
    int small = arr[0];
    for(i=0;i<n;i++){
        if(small > arr[i]){
            small = arr[i];
        }
    }
    return small;
}

int main(){
    int i, n, arr[10];
    int small;

    //Specify array total
    scanf("%d", &n);

    for(i=0;i<n;i++)
    {
        //Add numbers into the array
        scanf("%d", &arr[i]);
    }

    //Call Smallest function
    small = smallest(arr,n);

    printf("The Smallest number is: %d", small);
    return 0;
}
```

Problem 1: Array

Given an array ‘arr’, determine the second largest number in the array

Constraints:

- The first input is the length of the array
- The following inputs are the numbers in the array

Example 1:

|Input: 3 1 2 3
|Output: 2

Example 2:

|Input: 5 1 3 6 4 2
|Output: 4

```c
#include <stdio.h>
//Write a program to determine the second(2nd) largest number in the array
int largest(int arr[], int n){
    int i;
    int large = ?;
    …
}
…
int main(){
    int i, n, arr[10];
    int large, ?;
    …
    printf("%d", ?);
    return 0;
}
```

Problem 2: String

Given a string ‘s’, determine the length of the string

Constraints:

- Do not use the ‘strlen’ function.
- The input has to be one word.

Example 1:

|Input: HELLO
|Output: 5

Example 2:

|Input: GOODLUCK
|Output: 8

```c
#include <stdio.h>
//Write a program to determine length of a string
int strlength(char arr[]){
    int i = 0;
    ?
    return i;
}

int main(){
    int n;
    ? arr[10];
    …
    printf("%d",n);
}
```

Problem 3: Pointer

Given a pointer 'ptr', allocate space to corresponding amounts of numbers(1-9) into the pointer space

Constraints:

- You must use malloc() to allocate space for the pointer
- The Output must be taken from the pointed space

Example 1:

|Input: 3
|Output: 1 2 3

Example 2:

|Input: 5
|Output: 1 2 3 4 5

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

void fillptr(int *p, int i){
    ?
}

//Write a program to allocate space and fill it with numbers (1-9)
int main(){
    int i, n;
    scanf("%d",&n);
    int *ptr = malloc(n * sizeof(?));
    if(ptr == NULL){
        //Allocation failure
        return 1;
    }
    …
    free(ptr);
    ptr=NULL;
    return 0;
}
```