Functions

Question – 6 marks

A triangle is a valid triangle if sum of its two sides is greater than the third side. This rule must be true for all three sides. For example, if sides of a triangle are: a, b, and c, a+b must be greater than c, a+c must be greater than b, and b+c must be greater than a.
Consider the following definition of struct Triangle, and the prototype of function **find_invalid_triangle:**

```
struct Triangle{
    int a, b, c;
};

int find_invalid_triangle (const struct Triangle* arr, int n);
```

Now, complete the definition of the function in the following space, assuming **arr** points to an array of **struct Triangles** with **n** element. Your function definition should return - **1,** if all elements of the array contain a valid triangle. Otherwise, if there are one or more invalid triangles in the array, should return the index number of the **last occurrence** of an invalid triangle in array **arr.**
You may also assume value of n is always greater than or equal 1.
EXAMPLE: if **arr** points to an array that contains 3 invalid triangles in elements arr[2], arr[5] arr[7], it should return 7, which is the index number of the last occurrence.

IMPORTANT NOTES:
1. In your function definition you are NOT allowed to declare ANY local variable, anywhere.
   50% or more marks will be deducted if you declare any local variable within your function or control structures (loops or if statements).
2. Your solution must be efficient and avoid any unnecessary operations.

A Possible Solution:

```c
int find_invalid_triangle(const struct Triangle* arr, int n)
{

    while(n >0) {.

        if((arr[n-1].a + arr[n-1].b <= arr[n-1].c) ||
           (arr[n-1].a + arr[n-1].c < arr[n-1].b) ||
           (arr[n-1].b + arr[n-1].c < arr[n-1].a))

            return n -1;

        n--;
    }
    return -1;
}
```

Question – 7 marks

In the following space rewrite the complete definition of the function called: `make_mirrored_string`.

As partial definition of this function shows, it receives an argument called `char*s` that points to a valid c-string. Then, this function makes the string mirrored. For example: if `s` points to string `"ABC"`, it will be changed to `"ABCCBA"`, if string is `"Car"`, it will be changed to `"CarraC"`, if it is `"Apples"` it should be changed to `"ApplesselppA"`
You can assume that the original space that pointer `s` points to, is always large enough to append additional characters. This function must also return the string after it is changed.

Notes:
1. You are not allowed to declare any local variables other than i and j in this function or its control structures (loops or if statements); otherwise you will lose 50% of the question's mark.
2. You are not allowed to use any C library function.

For your information, here is the incomplete version of the function. Rewrite the complete definition of the function in the following space:

```
char* make_mirrored_string (char* s){

    int i, j;

    /* Students must rewrite the complete definition
       of this function in the following space.*/
}
```

A Possible Solution:

```
char* make_mirrored_string (char* s){
    int i, j;
    i = 0;

    while(s[i]) i++;

    j = i--;

    while(i>=0){
        s[j] = s[i];
        i--;
        j++;
    }

    s[j] = '\0';

    return s;
}
```

Question – 6 marks :

In mathematics, a Fibonacci integer sequence will have starting values of 0 and 1. After that, each value is the sum of the previous two values.
For example, here is the first 9 Fibonacci numbers:
```
        0, 1, 1, 2, 3, 5, 8, 13, 21, …
```
In the following box, write the definition of the function `is_fibonacci`.
```
int is_fibonacci(const int* x, int n);
// REQUIRES: n >= 3. Elements x[0] ... [n-1] exist.
// PROMISES: Return value is 1 if x[0] ... x[n-1] are the first n
// numbers of a Fibonacci integer sequence. Otherwise, the return
// value is 0.
```

Note: You are not allowed to use any local variable, anywhere within your function. You will lose 50% of the mark if you declare any variables within your function or within its control structures such as loops or if statements.

A Possible Solution:

```
int is_fibonacci_sequence(const int* x, int n){


  if(x[0] != 0 || x[1] != 1) return 0;


  x+=2;


  while (n-2){.
    if(x[0] != (x[-1] + x[-2])).
       return 0;


    x++;
    n--;
  }


  return 1;
}
```

Question (4 marks):

The following partial C program should open a text file called `input.txt`, with several integer number separated by whitespace, reads one integer number at a time, using `fscanf`, and prints each value into another text file (`output.txt`), until the end of the input file is, reached. The number in the output file should be separated by a tab ('\t').
You may assume that the data in the text file are all valid integers and that both files will open successfully without errors. **Your job is to fill the missing code within the while loop.**

```c
#include <stdio.h>
int main(void) {
    int number;
    FILE *fp1 = fopen("input.txt", "rt");
    FILE *fp2 = fopen("output.txt", "wt");

    while (1) {
     // Missing code
    }

    fclose(fp1);
    fclose(fp2);
    return 0;
}
```

In the following space rewrite your answer, the complete implementation of the while loop. You don't need to rewrite the rest of the program.

Note: You are not allowed to declare any other variable in this program. Marks will be deducted if you declare any other variables.

In the following space write your answer:

Possible Solution:

```c
    while (1) {.
        if(fscanf(fp1,"%d", &number) == EOF).
            break;
        fprintf(fp2,"%d\t", number);

    }
```

Short Answer Questions

Question (2 marks):

Write a macro with three arguments, called IN_RANGE, that produces a value of 1 if the value of the first argument is greater than or equal to the second argument and less than or equal to the third argument, and otherwise produces a value of 0.
Write your macro in the following space:

Possible Answers:

```
#define IN_RANGE(x, y, z) (x) >= (y) && (x) <= (z) ? 1 : 0
#define IN_RANGE(x, y, z) (x) >= (y) && (x) <= (z)
#define IN_RANGE(x, y, z) (x) >= (y) ? (x) <= (z) ? 1 : 0 : 0
```

Question (2maks):

Consider the following C program:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct String {
    char * string;
};

int main(){
    struct String s1, s2;
    s1.string = malloc(20);
    s2.string = malloc(10);
    strcpy(s1.string, "ABC");
    strcpy(s2.string, "XYZW");
    s1 = s2;
    free(s1.string);
    printf("%s\n", s1.string);
    free(s2.string);
    return 0;
}
```

This program is defective and will most likely crash due to a runtime error.

In the following space place clearly explain what is wrong at which point it will crash. There is no partial mark if the answer is not complete and clear.

Answer:

Primary point to be mentioned:
- **The problem with program is that it calls the function free twice to de-allocated the allocated memory, which string member of the struct String points to.**

Secondary points that could be also mentioned:
- **There will be memory leak when we say s1 = s2.**
- **The line with printf tries to display the value of a de-allocated memory space which is not logically a proper statement, but it will not cause a runtime error**

Question (2 marks):

consider the following code segment in C:

```
char* p = malloc(20);
strcpy(p, "ABC");
char* s = p++;
p +=2;
free(p);
```

Now, assuming that all required header files are included and the allocation of memory by malloc succeeds, what is still wrong with this code that may cause a runtime error?

Please explain very clearly why? And, how the issue can be fixed.

Note: Partial mark will not be awarded if the answer is unclear or incomplete.

Answer:

Function free is going to remove the memory from heap via pointer p that is already moved. The correct way to de-allocate the memory is to use pointer s by saying: free (s);

AR Diagrams:

Question - 6 marks

In the following space, make a memory diagram (AR diagram) for **point one**.

```
struct smaller
  {
    char s[3];
    int i;
};
struct larger {
    struct smaller
  sm;
    char t[4];
};

void foo(struct larger
*p)
{
  int *q;
  q = &(p-
  >sm.i);
  (*q)++;

  // point

  one

  return;
}

int main(void)
{
  struct larger x = {
      { "AB", 42 }, "CAN"
  };
  foo(&x);
  return
  0;
}
```
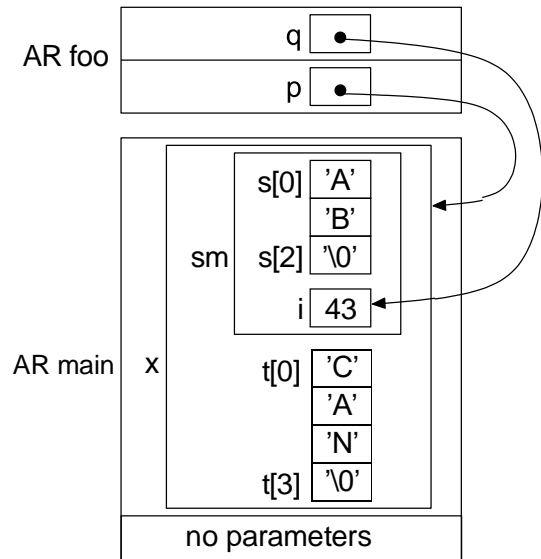
| AR foo | q | ● |
| | p | ● |

AR main | x |

sm	s[0]	'A'
		'B'
	s[2]	'\0'
	i	43
	t[0]	'C'
		'A'
		'N'
	t[3]	'\0'

no parameters

Question – 7 marks:

Draw an AR diagram for point ONE in the following program. Then, either insert the image of your diagram in the following box or use the "Add File" button to attach your file as your answer to this question.

```
long int function(char s[2], char p[1]){
    long int y = p - s;
    // POINT ONE
    return y;
}

int main() {
    int i = 0;
    const char* str1 = "ABC";
    char str2[] = "XY";
    char str3[4] ;
    str2[0] = *(str2 + 1);
    while(*str1){
        str3[i] = *str1 + 1;
        str1++;
        i++;
    }

    function(str3, str3 + i);
    return 0;
}
```


Question – 4 marks

Assume that the call to `malloc` succeeds. Draw a memory diagram for `point one` in the following space.

```c
#include <stdio.h>
#include <stdlib.h>
int glob[3] = { 31, 57 };
int main(void) {
  int loc[3];
  int *dyn =
    malloc(3 * sizeof(int));
  loc[0] = glob[0];
  dyn[1] = glob[1];

  // point one
  return 0;
}
```

Multiple Choice Questions

Question:

What is the output from the following code fragment?

```
char s[ ] = "university"; char *p = s;
int i;
for (i = 0; i < 4; i++) {
  (*p)++;
  fputc(*p, stdout);
  p++;
}
```

a) univ
b) nvri
c) uies
d) nive
e) vojw

Question:

Consider following structures and the main function:

```
typedef struct Info{
     char fname[20];
     char lname[30];
     int age;
}Info;

typedef struct Player{
     Info p_info;
     double shot_accuracy;
     double speed;
}Player;

int main ()
{
     Player flames [20];
     Player *p;
     p = flames;
}
```

With regard to above code, which one of the following lines of code can be used to change "age" for the fist index of the array flames?

a) (*p).p_info.age = 32;
b) p ->p_info.age = 32;
c) flames[0].p_info.age = 32;
d) (*flames).p_info.age = 32;
e) (*p)->p_info.age = 32;
f) p.p_info.age = 32;
g) flames->p_info.age = 32;
h) p.p_info->age = 32;

Question:

Consider the following small program:

```
void foo (int *x, int *y, int *z){
     *z = (*x) * (*y);
}

int main (){
     int a [5] = {10, 20, 30};
     int *b = a;
     int c = 5;

     //Function call goes here
}
```

a) foo (a[0], b[1], &c);
b) foo (a, b, &c);
c) foo (a[], b[], &c);
d) foo (&a, &b, &c);

Question:

Consider the following main function:
Assuming all required hearer file are included, what is the value of p at point one.

```
int main(){
    char *s = malloc(30);
    strcpy(s, "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"); // 29 Xs
    float* p = (float*)s;
    p+=2;
    // point one
    return 0;
}
```

a) &s[4]
b) s[8]
c) &s[8]
d) s[0]
e) &s[0]
f) s
g) s[4]

Question: Assuming size of pointer is 8 bytes, what is the output from the following code

```
char *a = "Banff"; char b[ ] = "Edmonton";
double c[ ] = {0, 3.141592653589793, 6.283185307179586};
int d[ ] = {sizeof(a), sizeof(b), sizeof(b[2]), sizeof(c)}; int i;
for (i = 0; i < 4; i++) printf("%d ", d[i]);
```

fragment?

a) 8 8 1 24

b) 8 9 1 24
c) 6 9 2 24
d) 8 9 1 3
e) 6 9 1 3

Question: Which line(s) in the following code fragment will cause a compilation error?

```
int a[4] = { 11, 22, 33 };    // line (1)
const int *b = a;             // line (2)
*b += 1;                      // line (3)
b += 2;                       // line (4)
```

a) Line (1)

b) Line (2)

c) Line (3)

d) Line (4)