

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)
ORGANISATION OF ISLAMIC COOPERATION (OIC)
Department of Computer Science and Engineering (CSE)

SEMESTER FINAL EXAMINATION

WINTER SEMESTER, 2016-2017

DURATION: 3 Hours

FULL MARKS: 150

CSE 4107: Structured Programming I

Programmable calculators are not allowed. Do not write anything on the question paper.

There are 8 (eight) questions. Answer any 6 (six) of them.

Figures in the right margin indicate marks.

- ✓ 1. Consider the following structure for a linked list:

```
struct student_marks {
    char roll[10];
    double marks;
    struct student_marks *next;
} *head;
```

One example of a linked list is given in Figure 1. In this example, there are 3 elements in the linked list. Your code should be able to handle any number of elements in the list. Write the following functions. Use the exact return type and parameters specified for each of the functions.

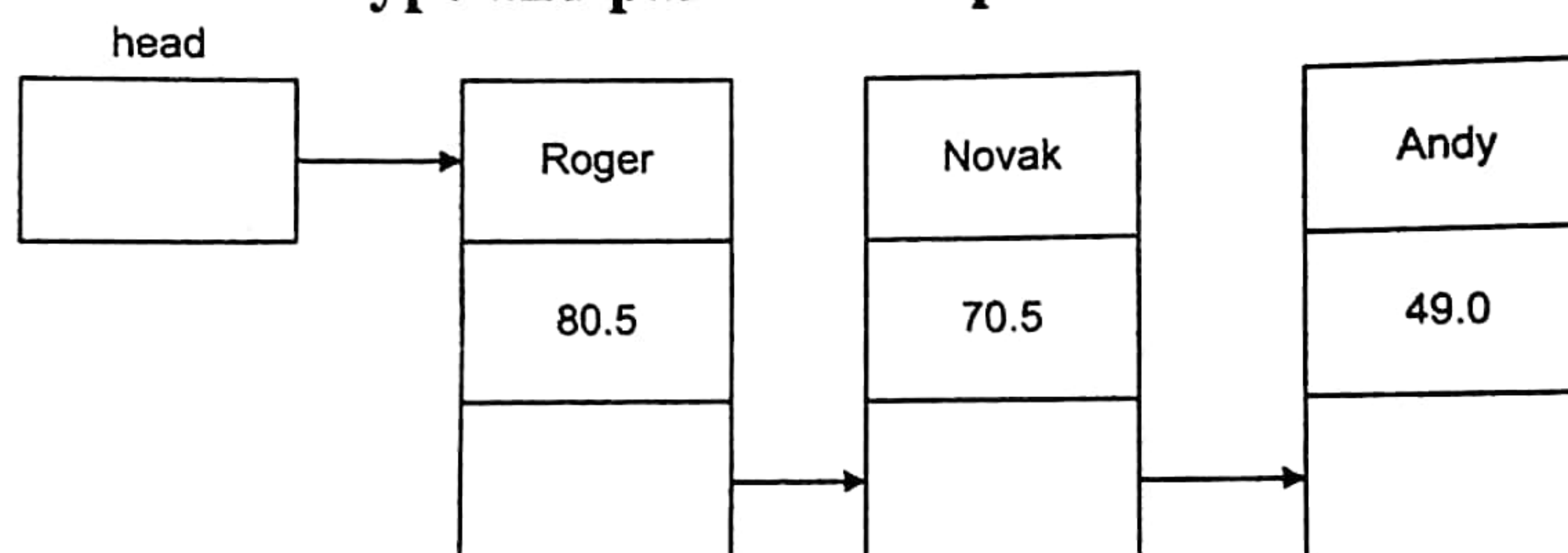


Figure 1: Linked list example for question no. 1

- a) Write the function **double getaverage()** that returns the average mark obtained by the students in the linked list. 10
 - b) Write the function **void getnames(double m)** that prints the name of the students whose marks are greater than **m**. 8
 - c) Write a function **double getlowest()** that returns the lowest marks obtained by a student. 7
2. a) Tapos is a student of IUT who is taking the Structured Programming I course. He decided to write down the marks obtained by each of the students in the first quiz of the course in a file. He wrote a C program that continues taking double type values as input from the console and writes them in a file called 'marks' in binary mode. After writing the marks of several students he decided to take a break. When he came back to his work he could not remember how many student's mark he had already written. So, he decided to write another program that can calculate the number of marks that has been written in the file. Help Tapos by writing a program that calculates the number of double type values written in the file and prints that number **in the console**. 7
- b) Differentiate between static memory allocation and dynamic memory allocation with the help of suitable examples. 8
 - c) How can the shift operators be used to perform multiplication and division by 4? Give suitable examples. 5
 - d) Write a program that takes two integer numbers as input and prints the result of the three binary bitwise operations performed on them. The program also prints the result of the unary operation performed on the first number. 5

3. a) The following codes in Figure 2, gives the output '210'. Rewrite the code so that the output is '211'. Do not change any of the printf() statements.

```
#include<stdio.h>
#define MAX(i, j) i > j ? i : j
int main(){
    printf("%d", MAX(1, 2));
    printf("%d", MAX(1, -1));
    printf("%d", MAX(100 && -1, 0));
    return 0;}
```

Figure 2: Code for question no. 3(a)

- b) The following two arrays are given which are already sorted in ascending order:

```
int A[10] = {2, 3, 3, 4, 5, 7, 9, 9, 12, 13};
int B[10] = {5, 5, 6, 8, 8, 9, 10, 12, 15, 17};
```

Write a program that will create a new array C[20] which will contain the elements of both A and B in ascending order.

- c) Write down the output of the code given in Figure 3.

```
#include <stdio.h>
int main(){
    int a[10]={10,9,8,7,6,5,4,3,2,1};
    int *p1,t,u,v,w,x,y,z;
    p1=a;
    t=(*p1)++; 10
    u=++(*p1); 12
    v=*p1; 12
    w=++*p1; 9
    x=*p1++; 9
    y=*(++p1); 7
    z=*(p1+1); 6
    printf("%d %d %d %d %d %d %d",t,u,v,w,x,y,z);
    return 0;}
```

Figure 3: Code for question no 3(c)

4. a) The Super-Secret Programmers Group (SSPG) is looking for new members and you are one of the lucky few who have the chance to join them. All you have to do is read what is written in their encrypted text file.

The encrypted text file contains a single line of **only characters** written in SSPG's own encryption method. In this method, the string that is to be written in the file is first reversed.

Next, the 1's complement of each of the characters of the string is written in the file.

So, in order to read the actual contents of the file you have to write a program that prints the 1's complement of each of the characters of the file in reverse order.

Example:

Suppose the string in the file is: abc

In the console you have to print, first the 1's complement of c, then the 1's complement of b and finally the 1's complement of a.

- b) Discuss the steps involved in solving a problem with the help of suitable examples.

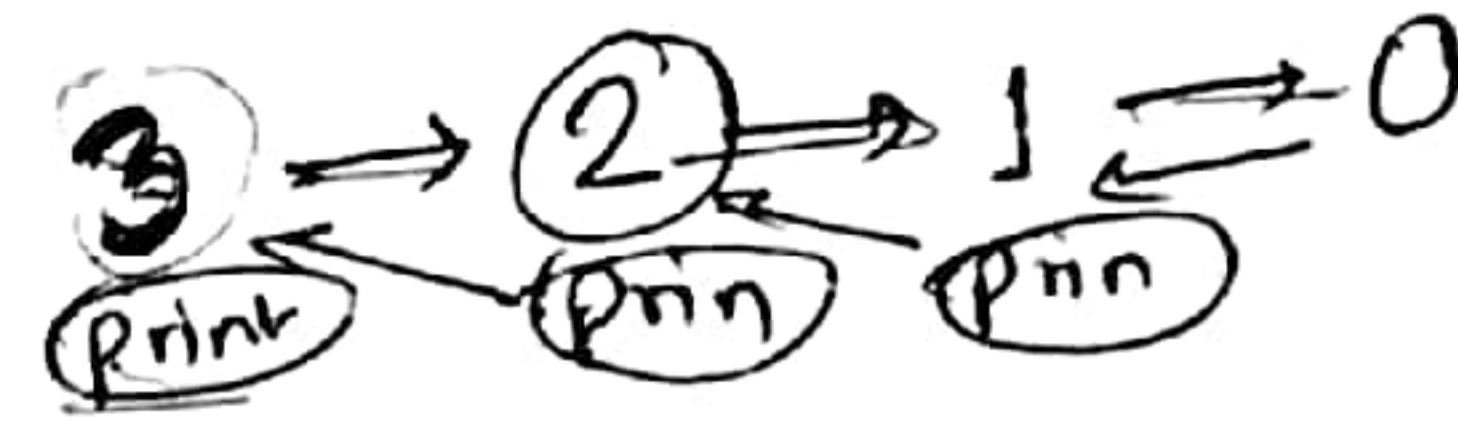
5. a) Farhan wrote a code that reads the text file 'names.txt' in which the name 5 different people are written in 5 separate lines. He knew the name of any person written in the file will have at most 13 characters. However, when he ran his code he did not get his desired output. The text file and the output of Farhan's code is given in Figure 4.

names.txt	Output
Elon Musk	Elon Musk
Sundar Pichai	Sundar Picha
Steve Wozniak	i
Steve Jobs	Steve Woznia
Jackie Chan	k
	Steve Jobs
	Jackie Chan

Figure 4: Input and the output obtained for question no. 5(a)

It can be seen that some of the names are written in two separate lines and there is additional new line between any two names. The code Farhan wrote is given in Figure 5.

```
#include<stdio.h>
int main(){
    FILE *fp;
    char name[14];
    fp = fopen("file.txt", "r");
    if(fp == NULL){
        printf("Error");
        exit(1);}
    while(1){
        if(fgets(name, 13, fp) == NULL)
            break;
        if(ferror(fp) != 0){
            printf("Error");
            exit(1);}
        puts(name);
    }fclose(fp);
}
```



1118

Figure 5: The code for question no. 5(a)

Your task is to help Farhan by pointing out the mistakes all the mistakes Farhan has made in his code. **You do not have to re-write the entire code correctly. You only have to point out and explain the errors in the code that led to the wrong output bring printed in the console.**

- b) What is the difference between the following two statements (if any)? Explain your answer. 6

```
int *arr1[8];
```

```
int *(arr2[8]);
```

Figure 6: Code for question no 5(b)

- c) Write a program that reads a very large number (at most 100 digits) from the user and counts the frequency of each digit and then prints that number of stars(*) for each digit. Your output should follow the format given below: 9

Input	Output
004528976736857325	0: ** 1: 2: ** 3: ** 4: * 5: *** 6: ** 7: *** 8: ** 9: *

Figure 7: Sample input and output for question 5(c)

6. a) Write a program which takes a character as input and prints the number of 0s and 1s that are present in the binary format of the ASCII code of that character. 12
- b) The Pascal's Triangle is an interesting number pattern which can consist of multiple lines. The only element in the first row is 1. To construct elements of the following rows, add numbers in its left and right from the previous row. If either the number to the right or left is not present, substitute a zero in its place. 13

The example given below is for the first 6 rows of the triangle:

				1				
			1		1			
		1		2		1		
	1		3		3		1	
1		5		10		10		5
	1		6		15		6	
		1		7		21		7
			1		8		28	
				1		9		36

[Here 1st element of 2nd row is 0 + 1 = 1
2nd element of 4th row is 1 + 2 = 3
3rd element of last row is 4 + 6 = 10]

Figure 8: Pascal Triangle for question no. 6(b)

Write a C program that will read a number N ($1 \leq N \leq 10$) from user. Then print first N rows of this triangle. Your output should be aligned properly so that it looks like a triangle.

- ✓ 7. a) The library function `strlen()` returns the length of the null-terminated string pointer passed to it. Write a recursive C program that performs the same task. The main function will take a character array as input and pass a **pointer to the array** to the recursive function. The recursive function will return length of the string pointer passed to it back to the main function. Print the returned value in the main function. **Do not use any extra array or print anything in the recursive function. Do not use the library function `strlen()` or any global variable.** 12
- b) Differentiate between the 'r', 'w', 'r+' and 'w+' modes of the `fopen()` function. 8
- c) The following two blocks of code tries to perform the same task. Is one of the code better than the other one? If so, which one and why? 5

```
int *i;
i = calloc(1, 4);
```

```
int *i;
i = calloc(1, sizeof(int));
```

Figure 9: Code for question no 6(c)

8. Zobaed is an avid follower of English Premier League Football and likes to collect the statistics of the players. While browsing through the internet he found many websites that record the goals and assists of the attacking players, but he did not find any good website which records the statistics of the defenders. So, he took it upon himself to create a record of the all the defenders of the league. He decided to arrange the information in the following manner:
- The structure **player** will contain the following information:
- Name of the player (Maximum 30 characters)
 - Number of matches played (Integer)
 - Number of blocks made (Integer)
 - Number of interceptions made (Integer)
- The structure **team** will contain the following information:
- Name of the team (Maximum 30 characters)
 - Position of the team in the league (Integer. Any value from 1 to 20. No two team will have the same value.)
 - Number of defenders in the team (Integer)
 - Array of type structure player (Assume no team has more than 10 defenders)
- a) Create the required structures according to the given specifications and create variables so that all the information can be stored. There are 20 teams in the league. 7
- b) Write a function that will print the player's name, team's name, number of blocks and number of interceptions made by each player. The players who play for the same team should be printed together. The players of the team with a better position (smaller number is better position) should be printed first. 10
- c) Write a function that will calculate the maximum number of matches played by any player and print the name of the all the players who have played the maximum number of matches. 8