



Beijing-Dublin International College



SPRING TRIMESTER MID-TERM EXAMINATION - (2022/2023)

School of Computer Science

COMP1002J Introduction to Programming 2

Dr. Robert Ross
Assoc. Prof. Neil Hurley
Dr. Brett Becker*
Dr. Seán Russell

Time Allowed: 60 minutes

Instructions for Candidates:

Answer all Questions.

BJUT Student ID:_____ UCD Student ID:_____

I have read and clearly understand the Examination Rules of both Beijing University of Technology and University College Dublin. I am aware of the Punishment for Violating the Rules of Beijing University of Technology and/or University College Dublin. I hereby promise to abide by the relevant rules and regulations by not giving or receiving any help during the exam. If caught violating the rules, I accept the punishment thereof.

Honesty Pledge:_____ (Signature)

Instructions for Invigilators

Non-programmable calculators are permitted.

No rough-work paper is to be provided for candidates.

Question 1:

a. Determine the output of the following code:

```

1 #include <stdio.h>
2
3 int main() {
4     int size = 5;
5     int doub = size * 2;
6     int a[doub];
7     int c = 0;
8     for (c; c < doub; c++) {
9         a[c] = c * c;
10    }
11
12    int i = size;
13    int temp = 0;
14    while (i < 10) {
15        temp = a[i - size];
16        a[i - size] = a[i];
17        a[i] = temp;
18        i = i + 1;
19    }
20    for (c = doub - 1; c >= 0; c--)
21        printf("%d ", a[c]);
22 }
```

$$a[0] = 0$$

$$a[1] = 1$$

$$a[2] = 4$$

$$a[3] = 9$$

$$a[4] = 16$$

$$a[5] = 25$$

$$a[6] = 36$$

$$a[7] = 49$$

$$a[8] = 64$$

$$a[9] = 81$$

$$a[1] = 36$$

$$a[2] = 49$$

$$a[3] = 64$$

$$a[4] = 81$$

$$a[5] = 100$$

$$a[6] = 121$$

$$a[7] = 144$$

$$a[8] = 169$$

$$a[9] = 196$$

Answer:

16 9 4 1 0 81 64 49 36 25

(12%)

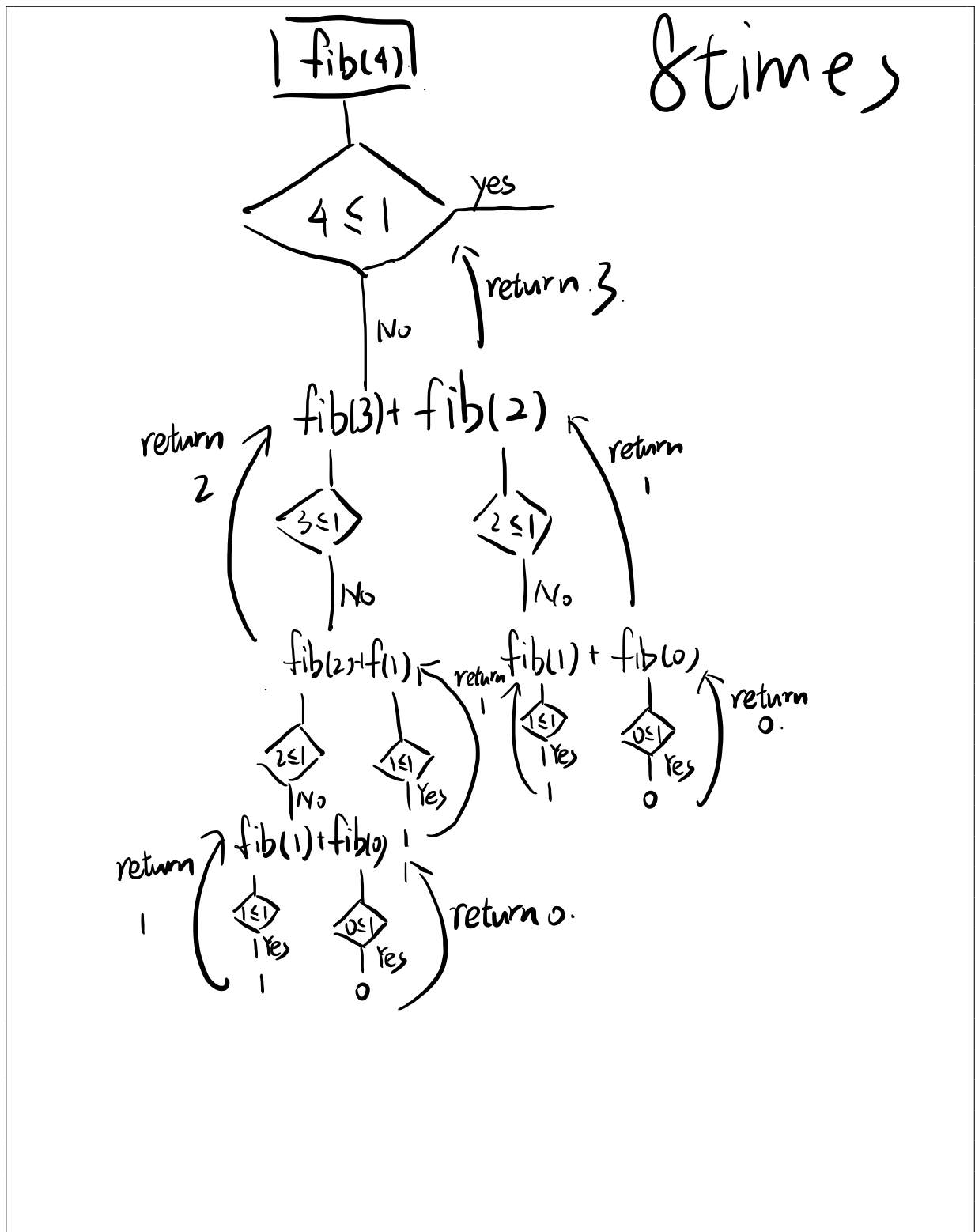
b. Consider the definition of the following recursive function for calculating the values in the Fibonacci sequence:

```

1 int fib(int n) {
2     if (n <= 1) {
3         return n;
4     } else {
5         return fib(n - 1) + fib(n - 2);
6     }
7 }
```

If the function is called using the code `fib(4)`, how many times in total is the function `fib` called? Draw a graph to show where each call to `fib` is made.

Answer:



(16%)

- c. What is the return type of the malloc function? Why is it not always necessary to cast the return value of malloc to the correct type?

Answer:

void*

void* can be cast to any data type
 because implicit type casting which is secretly
 using a type cast to change the type of (10%)
 the return point.

- d. Write a single line of code to dynamically allocate an array of integers of size 10.

Note: This code must work no matter which compiler is being used.

Answer:

```
int *int_array = malloc(10 * sizeof(int*))
```

(12%)

- e. What happens when dynamically allocated memory is not freed? How can this become a problem as our program runs for longer?

Answer:

will happen memory leak.
 will slow down computer and crash it.

(12%)

- f. Write a complete program that defines a type called Point. A Point is a structure that should contain:

- An integer representing the x-coordinate of the point
- An integer representing the y-coordinate of the point

The main function of the program should create a Point and assign it the coordinates (3, 4). The program should then print the coordinates of the point to the screen in the format (x, y).

Answer:

```
#include <stdio.h>
typedef struct {
    int x ;
    int y ;
} Point ;
int main ( ) {
    Point A = { 3 , 4 } ;
    printf ( "(%d , %d) " , A.x , A.y ) ;
}
```

(16%)

- g. As the result of an accident in file copying, the code of the following program has been scrambled and without indentation. Rewrite the code in the correct order with the correct indentation to produce a program reads an unknown number of integers from the a file specified by the user and outputs the sum of those numbers.

```
1 #include<stdio.h>
2 #include<string.h>
3 #include<stdlib.h>
4
5 int main(){
6     printf("File not opened for reading");
7 }
8 int l = strlen(file_name);
9 FILE* read = fopen(file_name, "r");
10 file_name[l-1] = '\\0';
11 int current, total = 0;
12 char file_name[100];
13 fclose(read);
14 if (read == NULL){
15     printf("The total of the numbers in the file is %d", total);
16     printf("Enter the name of the file:");
17     while ( (fscanf(read, "%d", &current)) == 1){
18         fgets(file_name, 100, stdin);
19     }
20     total = total + current;
21     exit(EXIT_FAILURE);
22 }
```

Answer:

(22%)
(Question Total 100%)