Answer questions in the spaces provided on the question sheets. All parts of questions 1,2 and 3 **MUST** be answered to receive any points. Write your name in **printed** letters. **Unreadable or unformatted answers will not be checked!**

1. Write a short code snippet (no includes, function definitions, just code) that:

Declares a static 1D array to store 50 ints, and initialize it with zeros:

1 _____
2 _____
3 _____
4 _____
5 _____

Declares a pointer to a double variable, stores an address of (previously declared) variable d and changes the value of d using the pointer

1 _____
2 _____
3 _____
4 _____
5 _____

Write a function header that returns a double value and accepts a 2D static array of integer values of size 30 rows by 20 columns

1 _____
2 _____
3 _____
4 _____
5 _____

2. Write a short code snippet (no includes, function definitions, just code) that:

Dynamically allocate memory to store 50 doubles. Initialize stored values with a value of 9.5. Deallocate the memory.

1 _____
2 _____
3 _____
4 _____
5 _____

Open a file data.dat for reading. Read the first line and store it in a buffer of 1000 characters.

1 _____
2 _____
3 _____
4 _____
5 _____

Declare a structure student. It should contain student first, last name and an id number.

1 _____
2 _____
3 _____
4 _____
5 _____

3. Write a **complete program**. The program should open a file data.dat, with the data structure shown, and read in the (x, y) coordinates to 2, dynamically allocated arrays of appropriate size. Print 10'th and 15'th elements.

n
$x_1\ y_1$
$x_2\ y_2$
...
$x_n\ y_n$

4. Write a **function** of an appropriate type that initializes a 1D array of n doubles with the following formula $x[i] = a \ e^{\frac{1}{i+1}} + \frac{1}{b}$, a and b should be passed as parameters.

```
1                                          11
2                                          12
3                                          13
4                                          14
5                                          15
6                                          16
7                                          17
8                                          18
9                                          19
10                                         20
```

5. Write a **function** of an appropriate type that for a 1D array of **n** doubles calculates and returns the average value of all the elements. Additionally the function should find the element nearest the average value and provide its **value** and **index**. No global variables are allowed.

```
1                                          11
2                                          12
3                                          13
4                                          14
5                                          15
6                                          16
7                                          17
8                                          18
9                                          19
10                                         20
```

6. Write a **complete program** using functions developed in questions 4 and 5. Do not rewrite the functions, just prototypes. The program should: 1. **Open a file of the same structure like Q3**. 2. **Read data from a file and store it in two dynamic arrays**. 3. **Initialize one of the arrays with function from Q4, the other should be filled with random numbers.** 4. **Use function 5 to calculate average values of both arrays.** 5. **Print indices of the values nearest the average values of both arrays.** The program must fit in the box below:

```
1                                          21
2                                          22
3                                          23
4                                          24
5                                          25
6                                          26
7                                          27
8                                          28
9                                          29
10                                         30
11                                         31
12                                         32
13                                         33
14                                         34
15                                         35
16                                         36
17                                         37
18                                         38
```