

GEBZE TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

CSE108 LAB #01 – Spring 2018

Write a complete program which calculates Body Mass Index (BMI) with the following functions.

Program Flow:

1. Greeting user
2. Get height in centimeters
3. Get weight in kilograms
4. Convert height from centimeters to meters
5. Calculate BMI
6. Print BMI information

Function: void welcome (void);

This function prints out the followings.

----- BMI Calculator -----

Body Mass Index tells someone whether he/she has a healthy weight for his/her height. Tell us your height and weight, then we are going to tell you about your BMI category!

Function: int getHeight(void);

This function prints a message and gets the height from the terminal/user as an integer.

A sample run of this function would be:

Enter height (in cm): 178

Function: double getInputDouble (void);

This function prints a message and gets the wieght from the terminal/user as a double.

A sample run of this function would be:

Enter weight (in kg): 67.5

Function: double calculateBMI (int height, double weight);

This function calculates and returns the body mass index. You must use “double pow(double x, double y)” function from the “math.h” library.

Body mass index can be calculated using: $BMI = \frac{\text{weight} [\text{kg}]}{\text{height}^2 [\text{m}^2]}$

GEBZE TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

Function: void printBMICategory (double bmi);

This function takes the bmi value as input and prints out the corresponding category using the table below.

| Category | BMI Range | |
|--------------------|-----------|-------|
| Underweight | 0 | 18.49 |
| Normal | 18.5 | 24.49 |
| Overweight | 24.5 | 29.99 |
| Obesity | 30 | >30 |

For example, for BMI value of 25.6 would print:

Category: Overweight

Function: void printInfo (int height, double weight);

This function prints height, weight, bmi and its category.

For example, printInfo(172,80.0) would print:

Height: 172 cm
Weight: 80.0 kg
BMI: 27.04 kg/m²
Category: Overweight

GEBZE TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING
CSE108 LAB #02 – Spring 2018

1. (25 points)Switch Statement:

Write a C program to create a simple calculator using switch statements. The program performs addition, subtraction, multiplication or division depending the input from user.

Output:

```
Enter an operator (+, -, *, /): -  
Enter two operands: 32.5  
12.4  
32.5 - 12.4 = 20.1
```

2. (25 points) If Else – Logic Statements:

Write a C program to input basic salary of an employee and calculate gross salary according to given conditions.

Basic Salary <= 10000 : HRA = 20% * basic_salary, DA = 80% * basic_salary

Basic Salary is between 10001 to 20000 : HRA = 25% * basic_salary, DA = 90% * basic_salary

Basic Salary >= 20001 : HRA = 30% basic_salary, DA = 95% basic_salary.

gross_salary = basic_salary + DA +HRA

Output:

```
Input  
Input basic salary of an employee: 22000  
Output  
Gross salary = 44000
```

3. (50 points) Loop Statements:

Write a c program :

- a. (20 points) Enter 10 numbers between 0 and 1000 from the keyboard. Find all the prime numbers within these 10 numbers and write them on the screen.
- b. (30 points) If there is not a prime number, write 'there is not a prime number' on the screen and then ask user to enter new 10 numbers. The program asks user to enter new 10 numbers and the program runs the same processes until a prime number is entered.

Output:

a. Enter 10 numbers between 0 and 1000:

2 3 4 17 98 103 120 144 191 281

The prime numbers : 2 3 17 103 191 281

b. Enter 10 numbers between 0 and 1000:

4 6 8 10 12 14 16 18 20 22

There is not a prime number

Enter 10 numbers between 0 and 1000:

4 6 8 10 12 14 16 144 191 281

The prime numbers : 191 281

CSE108 Lab04

Part 1 (25 points): Get a positive integer from user. At each line, sum up values starting from 1 to line number. Examples for n=1, 2 and 3 are illustrated below.

| | | |
|-----|--------------|-------------------------|
| 1=1 | 1=1 1+2=3 | 1=1 1+2=3 1+2+3=6 |
|-----|--------------|-------------------------|

Part 2 (25 points): Get a side length from user and draw the hexagon using nested for loops. An example hexagon for side length 1, 2 and 3 are illustrated below.

| | | |
|---|------------------------|---|
| * | 1 * *** *** * | 1 2 * 1 2 *** 1 ***** ***** ***** *** * |
|---|------------------------|---|

Part 3 (50 points): In blind hit game, your program puts a target to randomly determined distance between 5 and 10 and you can throw the ball 3 times. If you hit the target you win. Otherwise, you lose. To get a grade (25 points) from this part you must get a throwing distance from user and draw the ball projection shape as illustrated below.

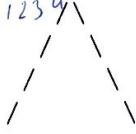
| Distance | Ball Projection Shape |
|----------|-----------------------|
| 1 | - |
| 2 | / \ |
| 3 | / \ \ |
| 4 | / \ / \ \ |
| 5 | / \ \ / \ \ \ |

How to generate random number: (Hint: To generate a random number between any two value (max-min) you can use modulus operator.)

| Code | Output |
|--|--|
| #include <stdio.h> #include <stdlib.h> #include <time.h> int main() { rand(time(NULL)); for (int i = 0; i < 5; i++) { printf("%d\n", rand()); } return 0; } | 1458004943 749093936 292808755 189749064 670949258 |

GEBZE TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

Example Program Output:

| | |
|--|--|
| Assume that program randomly determines target distance as 7. | Assume that program randomly determines target distance as 6. |
| Trial 1 - Enter throwing distance: 5  You could not hit the target! | Trial 1 - Enter throwing distance: 5  You could not hit the target! |
| Trial 2 - Enter throwing distance: 7  You hit the target! | Trial 2 - Enter throwing distance: 7  You could not hit the target! |
| Target distance is 7. | Trial 2 - Enter throwing distance: 8  You could not hit the target! |
| | Target distance is 6. |

Grading of Part 3:

- Only drawing ball projection (based on user input) is 25 points.
- Implementing the whole program with drawing dashed line (as many as distance) instead of ball projection gets 25 points.

Topic: Returning Multiple Parameters Functions in C

Do not use array!

Part 1 (30 points):

(30 points) Write a C program that uses LCMGCD() function returning both the greatest common divisor of 2 numbers and the Least Common Multiplier of 2 numbers together in one function call. Two integers are entered from the keyboard.

Numbers are 36 48
LCMGCD (36, 48)= 144, 12

$$\begin{array}{r} 36 \quad 48 \\ 18 \quad 24 \\ 18 \quad 12 \\ 18 \quad 6 \\ 18 \quad 3 \\ 12 \quad 3 \\ 6 \quad 3 \\ 3 \quad 1 \end{array}$$

✓

Part 2 (30 points):

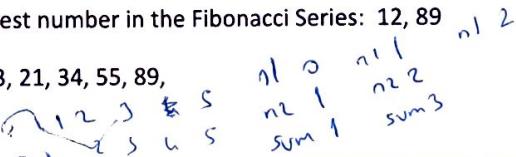
Write c program that finds Fibonacci Sequence Up to a Certain Number entered from the keyboard. Use Fibonacci() function to return number of the integers and the biggest number in the Fibonacci Series and display your Fibonacci Series.

The Fibonacci Sequence is the series of numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ... The next number is found by adding up the two numbers before it.

Enter number: 100

Number of the integer and the biggest number in the Fibonacci Series: 12, 89

Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89,



0112

Part 3 (40 points)

Write C program a sumofPrimeCheck() function to check an entered integer p from the keyboard. If an integer "p" is expressed as the sum of two prime numbers, calculate all possible combinations with the use of function, return the last a and b values. Display the all combinations on the screen. Else print "The number is not expressed as the sum of two prime numbers." on the screen.

Example output:

✓

Enter a positive integer = 34

All possible combinations:

34=3+31

34=5+29

34=11+23

34=17+17

Return: 17,17

GEBZE TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

CSE 108 Lab06

Topic: Using functions as argument, enumerated types and 1-Dimensional arrays

Part 1: Answer the question according to the scenario below.

Mehmet needs 51 cultural points to be successful in Cultural Activity course. Cultural points can be collected by three activities. Each activity has different cultural points and different ShowTime periods as shown in table below:

| Activity | Showtime Period | Cultural Point |
|----------|-----------------|----------------|
| Cinema | 5 | 1 |
| Theater | 7 | 3 |
| Concert | 10 | 2 |

*S, 2, 7
C
T
C
5
7
10
70
Friday*

For example; if the theater is showing today, next show will be seven days later. Lastly, Mehmet can only go to cultural activities on every five days.

- Calculate and print minimum number of days which Mehmet can obtain 51 cultural points (n days). You must have two functions:
 - int min_day_to_reach (int cultural_point) : Calculate minimum number of days which mentioned above.
 - int any_activity_point (int day_count) : Calculate how many cultural points obtained on given date.

Use any_activity_point function as an argument of min_day_to_reach function.

- Find the date after n days from today and print (Format example: 19/9/2019, Wednesday).

NOTES:

- * Today is 30/03/2018, Friday.
- * A month consists of 30 days.
- * The periods for all activities start from today.
- * Declare enumerated data type for days.

Output:

Mehmet can obtain 51 cultural points at least 150 days.

The date after 150 day from today is 30/08/2018, Monday.

Part 2:

Develop a program which get two positive integer numbers from user as one is size of array and the other is boundary of random numbers. The program creates a list and fill the list with random integer numbers between 0 and the boundary. Then the program calculates average of list items, finds closest item to the average value and stores it on actual_val.

Now the user asked to guess the actual_val in 5 times. In each time:

- If the user finds the value, congratulate and terminate program
- If guess is greater than actual_val and the difference is bigger than 10, then print "Cold, you should decrease your guess!" message
- If guess is less than actual_val and the difference is bigger than 10, then print "Cold, you should increase your guess!" message
- If the difference is less than 10, print "Hot, you almost find the average!" message.

Print a fail message and actual_val if the user can't find after five guesses and terminate the program.

Output:

Input:

| | |
|---|-----|
| Please enter size of array: | 100 |
| Please enter range of random numbers 0- to: | 250 |
| You have 5 guess to find average of the list. | |
| Make your guess (5 left): | 64 |
| Cold, you should increase your guess! | |
| Make your guess (4 left): | 76 |
| . | |
| . | |
| . | |

GEBZE TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

CSE108 LAB #07 - Spring 2018

In this lab assignment, you are going to implement a simple maze game. You must use enumeration to define wall, space, start (S) and finish (F) coordinates.

Maze01: (8x8)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| * | * | * | * | * | * | F | |
| * | | | * | | | * | |
| * | * | | * | | | * | |
| * | | S | * | | | * | |
| * | * | | | * | | * | |
| * | * | * | | * | | * | |
| * | | | | | | * | |
| * | * | * | * | * | * | * | * |

Maze02: (10x10)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| * | * | * | * | * | * | * | * | * | * |
| * | | S | * | | | * | | | * |
| * | | | * | | | * | | * | * |
| * | | | | * | | * | | * | * |
| * | | | | | * | | | * | * |
| * | | | | | * | | | * | * |
| * | | | | | * | | | * | F |
| * | | | | | * | | | * | * |
| * | | | | | * | | | * | * |
| * | * | * | * | * | * | * | * | * | * |

Function: void init_maze01 (...);

This function initializes Maze01 as illustrated above.

Function: void init_maze02 (...);

This function initializes Maze02 as illustrated above.

Function: void show_maze (...);

This function illustrates maze on terminal. You should put *, blank space, S and F for representing walls, empty slots, start and finish coordinates, respectively.

Function: void move (int maze[][], int pos_x, int pos_y);

This function checks if you can move up, down, left or right from the coordinate (pos_x, pos_y). If you are able do one of them, you do it. If you have more than one option, you choose one of them randomly. If you reach to the finish coordinate within 100 moves, you must print "Nothing like freedom!". Otherwise, you must print "Oh, trap!".

Function: int move_test (int maze[][], int pos_x, int pos_y, int direction);

This function checks if you can move up (1), down (2), left (3) or right (4) from the coordinate (pos_x, pos_y). If so, you must print "Test succeeded!". Otherwise, you must print "Test failed!".

Part 1 (100 points):

In this question you will be writing some functions to play the Sudoku game.

| | | | | | | | | |
|---|---|---|---|---|---|---|--|--|
| 5 | 3 | | 7 | | | | | |
| 6 | | | 1 | 9 | 5 | | | |
| 9 | 8 | | | | 6 | | | |
| 8 | | | 6 | | | 3 | | |
| 4 | | 8 | 3 | | | 1 | | |
| 7 | | | 2 | | | 6 | | |
| 6 | | | | | | | | |
| | 4 | 1 | 9 | | | 5 | | |
| | | 8 | | 7 | 9 | | | |

(a) Puzzle

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | 4 | 6 | 7 | 8 | 9 | 1 | 2 |
| 6 | 7 | 2 | 1 | 9 | 5 | 3 | 4 | 8 |
| 1 | 9 | 8 | 3 | 4 | 2 | 5 | 6 | 7 |
| 8 | 5 | 9 | 7 | 6 | 1 | 4 | 2 | 3 |
| 4 | 2 | 6 | 8 | 5 | 3 | 7 | 9 | 1 |
| 7 | 1 | 3 | 9 | 2 | 4 | 8 | 5 | 6 |
| 9 | 6 | 1 | 5 | 3 | 7 | 2 | 8 | 4 |
| 2 | 8 | 7 | 4 | 1 | 9 | 6 | 3 | 5 |
| 3 | 4 | 5 | 2 | 8 | 6 | 1 | 7 | 9 |

(b) Solution

0,0 0,1 0,2
1,0 1,1 1,2
2,0 ,2,1 2,2

- Sudoku is a 9x9 grid divided into smaller 3x3 boxes (also called regions or blocks) as shown in Figure 1(a).
- Some cells, called fixed cells, are populated with numbers from 1 to 9.
- The objective is to fill the empty cells, also called free cells, with numbers 1 to 9 so that every row, every column, and every 3x3 box contains the numbers 1 to 9 as shown in Figure 1 (b).
- Repeated numbers in each row, each column and every 3x3 box is not permitted.

void pretty_print(int grid[][9]): This function print the solution. Each row is entered as space seperated 9 integer numbers. Zero indicates an empty cell.

int isValid(int grid[][9]): This function checks whether the solution is a valid solution or not. Returns 1 if it is a valid solution and 0 otherwise. It checks if each row, column or block of the game is valid using the following functions.

int isValidRow(int grid[][9], int row): To be used by isValid, this function checks if the given row is valid and returns 1, 0 otherwise. A valid given row has all entries {1,2,3,4,5,6,7,8,9} exactly once.

Hint: Create an array of size 10 (call it digits) whose entries corresponds to a digit. Increase the corresponding array entry for each entry in the row. The initial array should be all zeros digits[10]={0,0,0,0,0,0,0,0,0}. If the row contains for example {1,2,3,4,4,5,6,7,8} then digits becomes {0,1,1,1,2,1,1,1,0}. Here digit 4 appeared in the row twice, and 0 and 9 never appeared. Once this array is formed you can decide if the row is valid. In other words, if the first entry is bigger than zero means invalid. For the rest, any entry not equal to 1 means invalid entry. Otherwise the entry is valid.

int isValidColumn(int grid[][9], int col): To be used by isValid, this function checks if the given column is valid. A valid given column has all entries {1,2,3,4,5,6,7,8,9} exactly once. Follow the same hint above.

int isValidBlock(int grid[][9], int i, int j): To be used by isValid, this function checks if the given block (i,j) is valid. The block (i,j) (for i=0,1,2 and j=0,1,2) has all entries { 1,2,3,4,5,6,7,8,9} exactly once.

Complete the following functions which are in the code. The code can be downloaded from Moddle:

(20 points) void pretty_print(int grid[][9]) ,(20 points) int is_valid_row(int grid[][9], int row) ,

(20 points) int is_valid_column(int grid[][9], int col) , **(20 points)** int is_valid_block(int grid[][9], int r, int c)

(20 points) int is_valid(int grid[][9])

A sample input and output of the program should be:

Enter a Sudoku puzzle solution (0 for blank cell):

```
5 3 0 0 7 0 0 0 0  
6 0 0 1 9 5 0 0 0  
0 9 8 0 0 0 0 6 0  
8 0 0 0 6 0 0 0 3  
4 0 0 8 0 3 0 0 1  
7 0 0 0 2 0 0 0 6  
0 6 0 0 0 0 0 0 0  
0 0 0 4 1 9 0 0 5  
0 0 0 0 8 0 0 7 9
```

Invalid solution

Another sample input and output of the program should be:

Enter a Sudoku puzzle solution (0 for blank cell):

```
5 3 4 6 7 8 9 1 2  
6 7 2 1 9 5 3 4 8  
1 9 8 3 4 2 5 6 7  
8 5 9 7 6 1 4 2 3  
4 2 6 8 5 3 7 9 1  
7 1 3 9 2 4 8 5 6  
9 6 1 5 3 7 2 8 4  
2 8 7 4 1 9 6 3 5  
3 4 5 2 8 6 1 7 9
```

Valid solution

Outputs:

Testing game in game001.txt.....

| | | |
|-------------------|-------------------|-------------------|
| 5 3 0 0 7 0 0 0 0 | 6 0 0 1 9 5 0 0 0 | 0 9 8 0 0 0 0 6 0 |
| 8 0 0 0 6 0 0 0 3 | 4 0 0 8 0 3 0 0 1 | 7 0 0 0 2 0 0 0 6 |
| 0 6 0 0 0 0 0 0 0 | 0 0 0 4 1 9 0 0 5 | 0 0 0 0 8 0 0 7 9 |
| 5 3 0 0 7 0 0 0 0 | 6 0 0 1 9 5 0 0 0 | 0 9 8 0 0 0 0 6 0 |
| 8 0 0 0 6 0 0 0 3 | 4 0 0 8 0 3 0 0 1 | 7 0 0 0 2 0 0 0 6 |
| 0 6 0 0 0 0 0 0 0 | 0 0 0 4 1 9 0 0 5 | 0 0 0 0 8 0 0 7 9 |

Invalid solution

Process exited after 1.613 seconds with return value 0
Press any key to continue . . .

Testing game in game002.txt.....

| | | |
|-------------------|-------------------|-------------------|
| 5 3 4 6 7 8 9 1 2 | 6 7 2 1 9 5 3 4 8 | 1 9 8 3 4 2 5 6 7 |
| 8 5 9 7 6 1 4 2 3 | 4 2 6 8 5 3 7 9 1 | 7 1 3 9 2 4 8 5 6 |
| 9 6 1 5 3 7 2 8 4 | 2 8 7 4 1 9 6 3 5 | 3 4 5 2 8 6 1 7 9 |
| 5 3 4 6 7 8 9 1 2 | 6 7 2 1 9 5 3 4 8 | 1 9 8 3 4 2 5 6 7 |
| 8 5 9 7 6 1 4 2 3 | 4 2 6 8 5 3 7 9 1 | 7 1 3 9 2 4 8 5 6 |
| 9 6 1 5 3 7 2 8 4 | 2 8 7 4 1 9 6 3 5 | 3 4 5 2 8 6 1 7 9 |

Valid solution

Process exited after 1.555 seconds with return value 0
Press any key to continue . . .

GEBZE TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

CSE 108 Lab10

Topic: Recursion

Part 1: Write the code for the function that is shown below for calculating least common multiple of two integer numbers recursively.

```
int lcm (int number_1, int number_2, int largest)
```

Input:

Enter two integer numbers for calculate the LCM of numbers: 24
36

Output:

LCM of 24 and 36 is 72.

24 36
| |
6 12
| |
3 6
| |
1 2

ali orr COZ
i read orre[3]

Part 2:

Write the code for the function that calculates number of vowels in given string recursively;

```
int count_of_vowels (const char *text, char *vowels, size_t vowels_count )
```

Input:

This is an example. Try to get numbers of vowels that are in the sentence.

Output:

Number of Vowels = 21

Part 3:

Write the code for the function that finds index of maximum item in an integer array recursively;

```
int max_of_array (const int *arr, size_t arr_size )
```

Input:

arr[10]=[12,1,162,523,22,621,76,36,57,235]

Output:

Index of maximum item in the array is 5 and the number is 621

GEBZE TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

CSE108 LAB #11 - Spring 2018

Question 1: Structure (50 points)

Write a C program using **structs** with the following grading policies:

- a) There are two quizzes, each graded on the basis of 10 points.
- b) There is one midterm exam and one final exam, each graded on the basis of 100 points.
- c) The final exam counts for 50% of the grade, the midterm counts for 25%, and the two quizzes together count for a total of 25% (do not forget to normalize the quiz scores).

The program will read in the students' scores and output the student's numeric score for the entire course. The program will display the average of the quizzes, tests, and the total score before it terminates.

A session with the program might look like this:

```
Enter the student's name: Mike Fisher
Enter the student's grade for quiz #1: 8      18
Enter the student's grade for quiz #2: 9      27
Enter the student's grade for midterm: 89      22
Enter the student's grade for final: 98      85
```

Mike Fisher's numeric score for the entire course is 92.5

Would you like to enter another student record? y(yes) or n(no)? y

```
Enter the student's name: Jon Stewart
Enter the student's grade for quiz #1: 7
Enter the student's grade for quiz #2: 8
Enter the student's grade for midterm: 78
Enter the student's grade for final: 84
```

Jon Stewart's numeric score for the entire course is 80.25

Would you like to enter another student record? y(yes) or n(no)? n

The class statistics:

```
The average score on quiz1 is 7.5
The average score on quiz2 is 8.5
The average score on midterm is 83.5
The average score on final is 91
The average score for the entire course is 86.375
```

GEBZE TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

Question 2: Recursion (25 points)

Write a C program using a recursive function to display the first n number (n is entered from the keyboard by the user). The numbers should be generated by the following formula:

$$f(n) = \begin{cases} 3, & \text{if } n == 1 \\ 5, & \text{if } n == 2 \\ f(n - 1) + f(n - 2), & \text{if } n \text{ is odd} \\ f(n - 1) - f(n - 2), & \text{if } n \text{ is even} \end{cases}$$

A session with the program might look like this:

Enter n = 12

```
f( 1 ) =      3
f( 2 ) =      5 )
f( 3 ) =      2
f( 4 ) =      7 )
f( 5 ) =      5 )
f( 6 ) =     12 )
f( 7 ) =      7
f( 8 ) =     19
f( 9 ) =     12
f( 10 ) =    31 )
f( 11 ) =    19 )
f( 12 ) =    50
```

Question 3: Union (25 points)

Write a union to store the roll no., name, age (between 11 to 14) and address of students (5 students). Store the information of the students.

- 1) Write a function to print the names of all the students having age 14.
- 2) Write another function to print the names of all the students having even roll no.
- 3) Write another function to display the details of the student whose roll no is given (i.e. roll no. entered by the user).

GEBZE TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

CSE108 LAB #12 - Spring 2018

In this lab assignment you are going to implement course grading system. In this system you must define a struct which stores the followings:

- Student Identifier [Union]: It is either student number or social number.
 - Student Number Given by university.
Format: year-department-no (3 digits) Example: 2018-CS-001
 - Social Number: Given by government.
Format: 11-digit number Example: 12345678901
- IdType: Indicator for type of student identifier. Its value is 0 and 1 for student number and social number, respectively.
- Name [10 Characters]
- Surname [10 Characters]
- Midterm 01 [Integer]
- Midterm 02 [Integer]
- Final [Integer]

Part1: In this part you must define student structure and create an array of students using `malloc()` where its size (how many students) is taken from user.

Part2: In this part you must write a function which creates a student. Name and surnames are taken from user. Midterm1, midterm2 and final are randomly assigned between 0-100.

Function: `void create (Student *student)`

Part3: In this part, you must write a function which shows information of a single student.

Function: `void print (Student student)`

Part4: In this part, you must calculate grades of all students. To do so, you must write a function which calculates grade of a single student. Weights of exams are 25%, 25% and 50% for midterm01, midterm02 and final exam.

Function: `void calc_grade (Student *student)`

Note: You can use `itoa()` function to convert an integer to a string.

Signature: `char * itoa (int src_value, char * output_str, int src_base);`

Example Code: This piece of code converts a 3 digit integer to a string.

```
int no = 123;
char buffer[3];
itoa(no,buffer,10);
```

GEBZE TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

CSE108 LAB #13 - Spring 2018

In this lab assignment you are going to implement a dynamic array.

Dynamic Array Structure:

- Pointer Array: It stores integer values. Whenever it is fully filled, its size grows by 5.
- Limit: Total size of array. It is initially 5.
- Current Size: Current size of array. It is initially 0.

Part1: ~~int~~ void add_to_last (Dyn_arr *arr)

This function initializes the array. All cells of the array is 0.

Part2: void add_to_last (Dyn_arr *arr, int input_value)

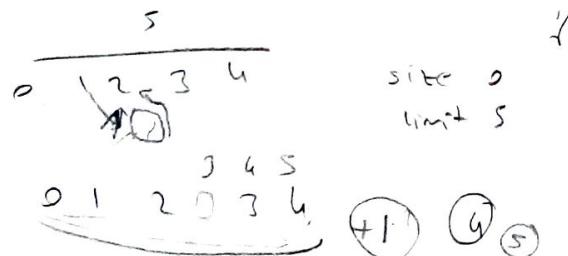
This function adds an input value to the next empty cell of array.

- Case 1: If array has one or more empty cells, this function simply add the input value to the end of array.
- Case 2: If array is full, this function enlarges array, shifts values and inserts the input value after index.

Part3: void insert_after (Dyn_arr *arr, int index, int input_value)

This function inserts input value after index on dynamic array.

- Case 1: If array has one or more empty cells, this function simply shifts values and inserts the input value after index.
- Case 2: If array is full, this function enlarges the array, shifts its values and inserts the input value after index.



CSE 108 - Spring 2018

Lab 14

Question 1:

For given a linked list, write a C programming function which deletes all the nodes from the list that are greater than the specified value x which is entered by the user.

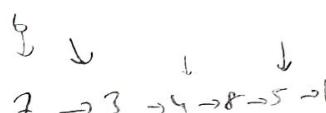
Use a linked list node:

```
typedef struct node {
```

```
    int val;
```

```
    struct node * next;
```

```
} node_t
```



Output:

```
Input : list: 7->3->4->8->5->1
```

```
    x = 6
```

```
Output : 3->4->5->1
```

```
Input : list: 1->8->7->3->7->10
```

```
    x = 7
```

```
Output : 1->7->3->7
```

Question 2:

Write a C program which writes, reads and updates a binary file (data.bin).

Binary file keeps track of a customer's first name, last name, work year, salary and account number (at most 4 digits, check it!).

a) fill (* struct, size, count)

This function creates a customer structure array with size count and gets values of each customer from user.

- The first argument is the structure.

- The second argument is the byte size of that structure.

- The third argument is how many of those structures to write.

b) Record the content of customer structure array into the file data.bin.

c) Write a print function to read the content of data.bin file.

fread (* struct, size, count, file)

- d) Update customers' payments in the file.
If work year is less than 5 years
 payment = payment + payment %20
Else
 payment = payment + payment%10.

Example:

First Name: Steve
Last Name : Dannelly
Acct Num : 1234
Work year: 5
Payment : 10000\$

First Name: Bob
Last Name : Jones
Acct Num : 321
Work year: 11
Payment : 15000\$

First Name: Sally
Last Name : Smith
Acct Num : 567
Work year: 4
Payment : 5000\$

Updated!

| | | | | | |
|--------|-------|----------|------------|------|--------------------|
| Name = | Steve | Dannelly | Acct Num = | 1234 | Payment = 11000 \$ |
| Name = | Bob | Jones | Acct Num = | 321 | Payment= 11500 \$ |
| Name = | Sally | Smith | Acct Num = | 567 | Payment = 6000 \$ |

GEBZE TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

CSE108 FINAL LAB - Spring 2018

Part1: void replace_word (char* fname, char* wsrc, char* wdest)

This function replaces the source word "wsrc" with the destination word "wdest" in content of a file "fname". To do so, you must read content of the file, do the replacement(s), and then write the content back.

Part2:

Part2-a) Using Structure:

A complex number is a number of the form $a + bi$, where a and b are real numbers and i is an indeterminate satisfying $i^2 = -1$.

For example, $2 + 3i$ is a complex number.

Addition and subtraction:

$$(a + bi) + (c + di) = (a + c) + (b + d)i.$$



$$(a + bi) - (c + di) = (a - c) + (b - d)i.$$

Multiplication and division:

$$(a + bi)(c + di) = (ac - bd) + (bc + ad)i.$$

$$i^2 = i \times i = -1.$$

Write a C program, struct two complex numbers which are entered by users. Your program calculates addition, subtraction multiplication or division of these two complex numbers according to chosen operation. Show your result on the console.

Example Output:

```
For 1st complex number
Enter real and imaginary part respectively: 2.3
4.5
```

```
For 2nd complex number
Enter real and imaginary part respectively: 3.4
5
Choose your operator: +
```

```
Sum of these two complex number = 5.7 + 9.5i
```

Part2-b) Using Linked-List:

Significant Number: If the sum of the all positive divisors less than the number is equal to the number, it is a significant number.

For instance: $28 = 1 + 2 + 4 + 7 + 14$, 28 is a significant number.

a) Calculate all significant numbers between 0 and 1000.

b) Show all divisors of them and sort them all together decreasing order.

GEBZE TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

Part3:

a-) Write the code of the function that finds and returns super digit of given integer recursively.

*Int super_digit(int number, struct Node** head_of_list)*

The super digit of an integer x is defined with following rules:

- If x has only 1 digit, then its super digit is x .
- Otherwise, the super digit of x is equal to the super digit of the sum of the digits of x .

For example, the super digit of 567829 will be calculated as:

$$\text{super_digit}(567829) = 5+6+7+8+2+9$$

$$= 37$$

$$\text{super_digit}(37) = 3+7$$

$$= 10$$

$$\text{super_digit}(10) = 1+0 = 1$$

The super digit of 567829 is 1. Print the result after you got it from the function.

input:

987697896

output:

The super digit of the 987697896 is 6

b-) Construct a linked list before continuing to part b. Write the code of the function that adds an integer to the end of the list. Call the *add_to_list()* function in *super_digit()* function to add integers that are gained from calculating the super digit of an integer to the linked list.

*int add_to_list(struct Node** head_of_list, int data)*

For the example that is shown in part a:

$$\text{super_digit}(567829) = 567829 \rightarrow \text{add } 567829 \text{ to list}$$

$$= 5+6+7+8+2+9$$

$$= 37 \rightarrow \text{add } 37 \text{ to list}$$

$$567829 \rightarrow 37 \rightarrow 10 \rightarrow 1$$

c-) Write the code of the function that prints reverse of the linked list which is constructed in part b recursively. Note that; the list won't be reversed, only print the list from tail to head.

input:

The linked list: 567829 -> 37 -> 10 -> 1

output:

1 -> 10 -> 37 -> 567829