

ASSIGNMENT BRIEF

HTU Course No: 30201100

HTU Course Name: Fundamentals of Computing

BTEC UNIT No:

BTEC UNIT Name:

Version: 2



Assignment Brief

Student Name/ID Number/Section	
HTU Course Number and Title	30201100: Fundamentals of Computing
BTEC Unit Number and Title	Not Available
Academic Year	Fall 2021/2022
Assignment Author	Eng. Malek Al-Louzi and Eng. Umar AlKafaween
Course Tutor	Dr. Nayef Abu Aqeel, Dr. Rami Al-Ouran, Dr. Hala Hamadah, Eng. Umar AlKafaween, Eng. Dania Alsa`id, Eng. Alaa Al-Bakhit, Eng. Hebah AlDahoud, Eng. Yazan AlMasri, Eng. Moath Abedalqader, Eng. Malek Al-Louzi
Assignment Title	Guessing Games
Assignment Ref No	Assignment 1
Issue Date	19 th December 2021
Formative Assessment dates	9 th January 2022 to 17 th January 2022
Submission Date	22 nd January 2022 at 23:00
IV Name & Date	Eng. Malek Al-Louzi, Eng. Dania Alsa`id and Eng. Hebah AlDahoud

Submission Format

This assignment is project design. You must complete it and submit according to the following guidelines (Failing to follow these guidelines may result an 'unclassified' grade):

Project Design Guidelines

- You are required to submit a well formatted report providing complete answers to all tasks as follows:
 - 1) A word or PDF file (use the format that was sent to you by the instructor) include:
 - a) **The flow chart** you used to implement your program, the **block diagram** with your justifications of **Task Six. (Don't use auto generated flow charts).**
 - b) The **student declaration form** attached to this assignment brief (use electronic signature to sign it).
 - 2) You must submit a **Source File** for the program you developed (extension should be .c like *YourName_FOC.c*). The source file should be **fully commented**. All your work of coding and compilation should be done using **Ubuntu and gcc** compiler only.
- Soft-copy submissions are only allowed, you are required to upload your submission files to the university's eLearning system through (<https://elearning.htu.edu.jo/>) within the submission date and time stated above. **NO SUBMISSION by EMAIL and NO LATE SUBMISSIONS WILL BE ACCEPTED.** Failing to follow this guideline may result in an 'unclassified' grade.

- If you commit any kind of plagiarism, HTU policies and regulations will be applied.
- This is a strictly individual assignment and no collaboration amongst students is allowed, where working with your colleagues is not a teamwork, it is a kind of plagiarism.

In-Class (Oral) part

- In-class assessment will be one to one **online** or **Face-to-Face** oral discussion between you and your instructor on Microsoft Teams or in the class, which includes designing, developing, debugging, analyzing, and evaluating the code and algorithm developed in this project.
- The attendance of the oral assessment is mandatory in the date and time determined by your instructor and be ready to **open your camera** (if it was an online discussion) from beginning of the assessment.
- Answering all tasks in the report with the code will be evaluated through the oral assessment. So, any question in the oral assessment for any criteria is not answered in the required level that mean you will lose those criteria, even though you answered it in the report and not included directly in a specific question.
- You must sign the witness form that your instructor will fill out during the discussion to complete the oral assessment process.

Unit Learning Outcomes

LO1 Discuss the basic concepts of computer hardware, software and operating systems, with the basic relation between them, taking C commands as an example.

LO2 Implement a full program in C with identification of the three kinds of control structures: sequence, selection, and repetition, understanding the capabilities of implementing C codes that capable to deal with functions.

LO3 Understanding the basic concepts of pointers in C with implementation of arrays and dealing with files. In addition to the role of C programming as a powerful tool to communicate between devices and manipulate their data.

Assignment Brief and Guidance

Scenario

You applied for an internship at a video game development company called Mintendo that is famous for producing small games for kids. As part of the hiring process, you are given a take-home assignment. The assignment is broken down into 6 tasks that you need to complete in order to be considered for the internship.

Task One

Implement separate flow charts for **Task Two** and **Task Three** of this Assignment, for documentation process issues. You can draw the flowchart on paper and take pictures of it, but you need to be extra careful to make sure the drawing and the pictures are clear. Alternatively, you can use any computer program to build the flowchart using basic shapes (e.g., draw.io) or online free website like (<https://app.diagrams.net/>).

Task Two

Write a C program that upon starting asks the user for their name, then displays a welcome message and prompts the user to choose one of 3 options:

1. to play a game called Guess The Number
2. to play a game called Hangman
3. to exit the program

The output of this task is shown in Figure 1.

```
Welcome Sanad!  
Please enter the number of the game you wish to play, or choose Exit.  
1. Guess the Number.  
2. Hangman.  
3. Exit.
```

Figure 1

- If the user enters 1, the program should run a function called **startGuessTheNumberGame** that just prints "Guess the Number has started."
- If the user enters 2, the program should run a function called **startHangmanGame** that just prints "Hangman has started."
- If the user chooses 3, the program should exit.
- If the user enters any other number, the program should display an error message "Invalid choice."

As long as the user does not enter 3, the program should carry out the appropriate action and display the menu again. The program will only exit if the user enters 3.

Task 3

Modify the program you wrote in Task 1 to update the function **startGuessTheNumberGame** to implement Guess the Number game, which works as follows:

The program asks the user to enter a secret integer between 1 and 10000 as shown in Figure 2.

```
Enter a secret number between 1 and 10000:
```

Figure 2

Then program will try to guess the number and print the guess on the screen.

After that the user will need to select one of 3 options available; confirm to the program that the guess is correct, or that the secret number is larger than the guess, or that the secret number is smaller than the guess as shown in Figure 3.

```
Enter a secret number between 1 and 10000: 350  
My guess is: 5000  
Please choose one of the following 3 options:  
1. My guess is correct.  
2. The secret number is larger than 5000.  
3. The secret number is smaller than 5000.
```

→ The user entered 350

Figure 3

- If the user chooses 1, the game ends, and the program will show to the users how many guesses did it take to find their secret number as explained in **task 4**, then it will return to the main menu.
- If the user chooses 2 or 3, the program will make another guess and display the 3 options again, and so on until it finds the secret number.
- If the user enters any other number, the program will display an error message "Invalid choice." and redisplay the guess and wait for a valid choice.

To implement this game, you are going to implement an algorithm called Binary Search. Binary Search is an algorithm that can search for a number among a group of sorted numbers in a very fast manner.

Consider the following example that searches among the numbers 1 to 8. Assume 3 is the secret number we want to find as shown in Figure 4.

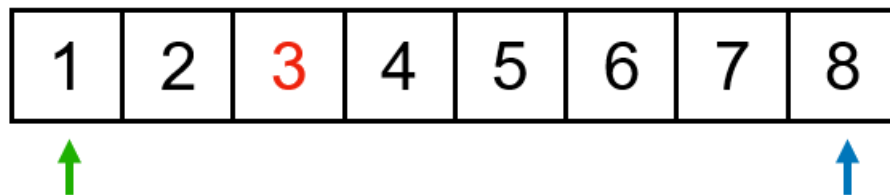


Figure 4

- First, the program will guess the middle number in the range 1-8, in this case 4.
- Then, the user will respond to the program that their secret number is less than 4.
- Now that the program knows that the secret number is less than 4, it will refine its search range to exclude all the numbers greater than or equal to 4. This results in discarding half the numbers. The new range will be 1-3 as shown in Figure 5.

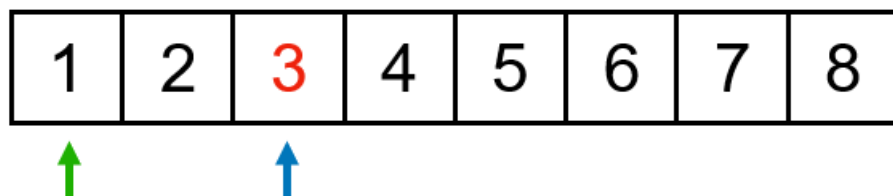


Figure 5

- Now, the program will make another guess, similar to the previous one, it will guess the

middle number in the range, in this case 2.

- The user will respond to the program that their secret number is greater than 2.
- Now that the program knows that the secret number is greater than 2, it will refine its search range to exclude all the numbers less than or equal to 2.
- This results in discarding half the numbers. The new range will be 3-8 as shown in Figure 6.

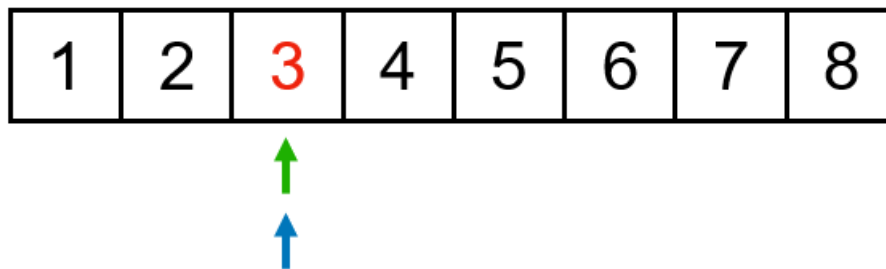


Figure 6

Repeating the steps again:

- The program will make another guess. Since the range contains one number only it will be the guess, in this case 3.
- The user will respond to the program that the guess is correct.
- The program will print to the user the number of guesses, which is 3 (first guess was 4, second was 2, and the last one was 3 - so 3 in total).

The following pseudo code describes the binary search algorithm:

```

step 1: START
step 2: secretNumber = 3
step 3: low = 1, high = 8
step 4: repeat the following as long as low ≤ high:
step 5:     mid = (low + high) / 2
step 6:     if (secretNumber == mid):
step 7:         print "Secret number found!"
step 8:     else if (secretNumber > mid):
step 9:         low = mid + 1
step 10:    else
step 11:        high = mid - 1
step 12: END
  
```


Task Four

After finishing the Task 2, you should show to the users how many guesses did it take to find their secret number using a column in an image, the dimension of the image and the place of the column are shown in Figure 7.

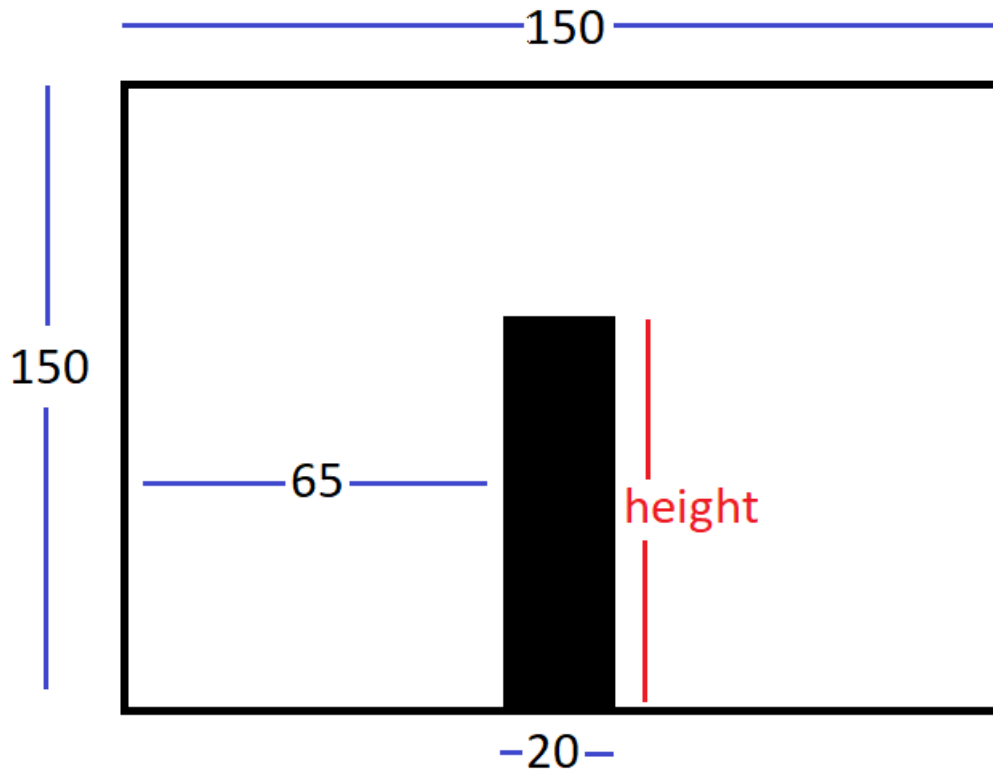


Figure 7

Note that the **height** of the column will reflect the number of guesses, each guess will be equivalent to exactly 10 units of height. For example, if the number of guesses is 3 the height of the column will be 30, if the number of guesses is 5, the height of the column will 50, and so on.

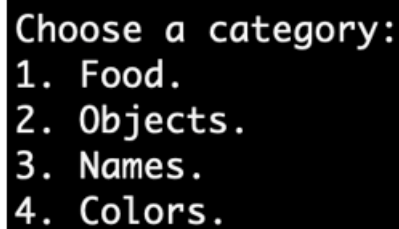
Task Five

Modify the program you have to update the function startHangmanGame to implement the Hangman game, which works as follows:

The program asks the user to select a category from one of the following:

1. Food.
2. Objects.
3. Names.
4. Colors.

As shown in Figure 8.

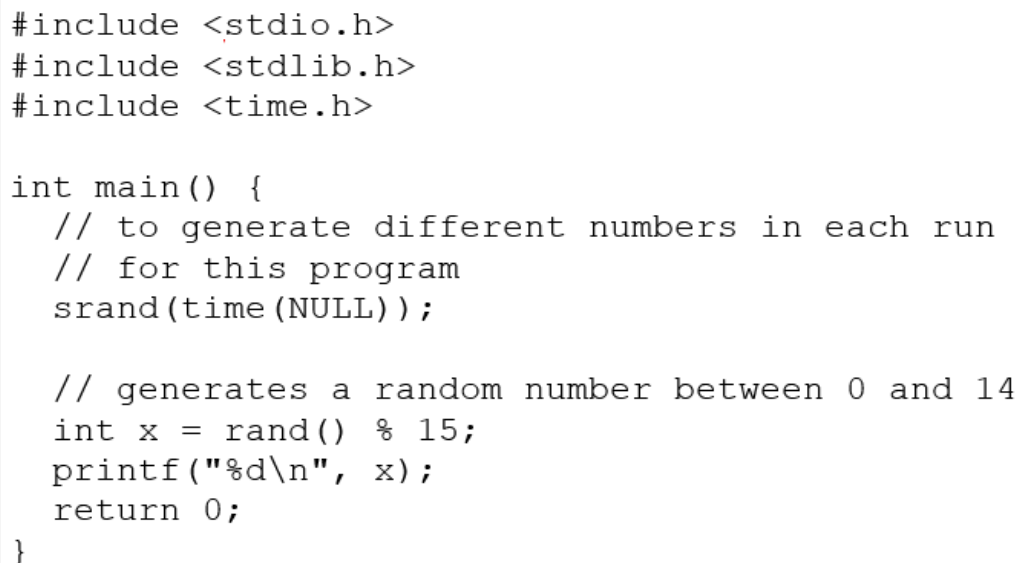


```
Choose a category:
1. Food.
2. Objects.
3. Names.
4. Colors.
```

Figure 8

- If the user enters one of the 4 numbers, the program will select a word from that category to play. If the user enters any other number, the program will display an error message "Invalid choice." and redisplay the categories and wait for a valid choice.
- The program will use the provided files [food.txt, objects.txt, names.txt, colors.txt] which contain 15 words each to select a word from the user-selected category.
- The selected word should be chosen in random.

You can use the rand function to generate random numbers in C, here is an example:

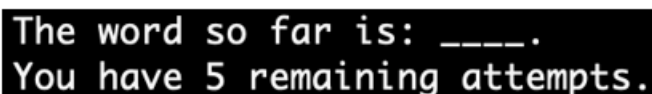


```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main() {
    // to generate different numbers in each run
    // for this program
    srand(time(NULL));

    // generates a random number between 0 and 14
    int x = rand() % 15;
    printf("%d\n", x);
    return 0;
}
```

After the random word is chosen from the file that corresponds to the chosen category, the program will display underscores that correspond to the hidden word characters, along with the remaining attempts as shown in Figure 9.



```
The word so far is: _____.
You have 5 remaining attempts.
```

Figure 9

- The word in the Figure 9 above contains 4 letters only, and the user has 5 attempts.
- Then, the user enters a letter, the program checks whether this letter exists in the word or not. If it does not exist, the program will deduct 1 from the remaining attempt and redisplay the previous message again as shown on Figure 10.

```
p
Not quite! Try again!
The word so far is: _____.
You have 4 remaining attempts.
```

Figure 10

- If the character exists in the word, the program will replace the underscore in that character's position with the character and redisplay the previous message as shown in Figure 11.

```
i
Correct! Keep going!
The word so far is: ___i.
You have 4 remaining attempts.
```

Figure 11

- If the number of attempts gets to zero, a hard luck message will be printed to the user as shown in Figure 12 and the game will exit to the main menu.

```
a
Hard luck! You were not able to find the word. :(
```

Figure 12

- If the user finds all the letters before running out of attempts, a congratulations message will be printed to the user as shown in Figure 13 and the game will exit to the main menu.

```
Congratulations! You found the word!
```

Figure 13

Task Six

This Code should be prepared to send data to another computer for statistical purposes, suppose your computer is at point A and the another computer is at point B. You should choose a communication protocol to connect point A to point B, note that this is not about writing a code, it's about selecting a physical connection. Identify the available communication protocols that can be used to implement the connection, select one of them and justify your selection. Use a block diagram to support your answer.

STUDENT ASSESSMENT SUBMISSION AND DECLARATION

When submitting evidence for assessment, each student must sign a declaration confirming that the work is their own.

Student name:		Assessor name:	
Student ID:			
Issue date:	Submission date:	Submitted on:	
	/ /2021	/ /2021	
Programme:			
HTU Course Name: Fundamentals of Computing BTEC UNIT Title *: NA			
HTU Course Code: 30201100		BTEC UNIT Code: NA	
I AM REPEATING THIS UNIT*:		(YES) (NO)	

Plagiarism

Plagiarism is a particular form of cheating. Plagiarism must be avoided at all costs and students who break the rules, however innocently, may be penalised. It is your responsibility to ensure that you understand **correct referencing practices**. As a university level student, you are expected to use appropriate references throughout and keep carefully detailed notes of all your sources of materials for material you have used in your work, including any material downloaded from the Internet. Please consult the relevant unit lecturer or your course tutor if you need any further advice

Student declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student signature:

Date:

Learning Outcomes and Assessment Criteria			
Learning Outcome	Pass	Merit	Distinction
LO1: Discuss the basic concepts of computer hardware, software and operating systems, with the basic relation between them, taking C commands as an example.	<p>P1 Identify the basic hardware of the computer and how to deal with it.</p> <p>P2 Identify the basic software of the computer, and able to interact with Ubuntu operating system through its terminal.</p>	<p>M1 Analyse the basic structure of simple C program.</p> <p>M2 Express any arithmetic or logical expression to its equivalent in C language.</p> <p>M3 Apply C language programming basic concepts to trace and debug simple C programs.</p>	D1 Critically analyse basic C commands and relate it with the machine cycle.
LO2: Implement a full program in C with identification of the three kinds of control structures: sequence, selection, and repetition, understanding the capabilities of implementing C codes that capable to deal with functions.	<p>P3 Implement problem solving decisions in algorithms and introduce structure charts as a system documentation tool.</p> <p>P4 Identify the C syntax of sequence, selection, and repetition and when to use each statement type.</p> <p>P5 Identify the structure of functions with passing information and get data if needed.</p>	M4 Solve real life problems using control structures in C language integrated with functions.	<p>LO 2 & 3</p> <p>D2 Critically evaluate the source code that solve real life problem implementing the functions, arrays, and files following the code standards and best practices.</p>
LO3: Understanding the basic concepts of pointers in C with implementation of arrays and dealing with files. In addition to the role of C programming as a powerful tool to communicate between devices and manipulate their data.	<p>P6 Explore pointer's role in referencing variables, and how it is reflected in the memory.</p> <p>P7 Implement arrays as one dimensional or two dimensional and dealing with any operation on them.</p> <p>P8 Explore the fundamentals of data communication between devices.</p>	<p>M5 Implement the effectiveness of arrays in real applications with usage of functions.</p> <p>M6 Identify streams in C and their relationship to deal with files.</p> <p>M7 Introduce a block diagram to solve a real-life problem in communication between devices as a practical application of C language.</p>	