Exercise: 1

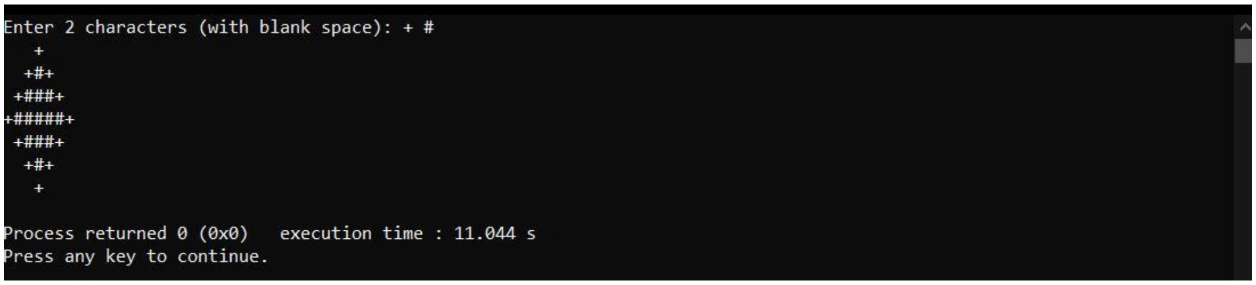
Part I: Source code

```
#include <stdio.h>

#include <stdlib.h>


// C functions cannot return 2 values, and we need two user inputs.

// So it's better to keep the variables in main() and call them by reference to store the user
input.

void info(char *character1, char *character2){

    printf("Enter 2 characters (with blank space): ");

    scanf("%c %c", character1, character2);

}


// function to print the diamond shape according to user inputs.

void diamond(char character1, char character2){

    printf("   %c   \n", character1);

    printf("  %c%c%c  \n", character1, character2, character1);

    printf(" %c%c%c%c%c \n", character1, character2, character2, character2, character1);

    printf("%c%c%c%c%c%c%c\n", character1, character2, character2, character2, character2,
character2, character1);

    printf(" %c%c%c%c%c \n", character1, character2, character2, character2, character1);

    printf("  %c%c%c  \n", character1, character2, character1);

    printf("   %c   \n", character1);

}


int main() {

    char character1, character2;


    // calling a function info with call by reference

    info(&character1, &character2);

    // calling function diamond with characters as parameters

    diamond(character1, character2);


    return 0;

}
```

Part 2: Software development process

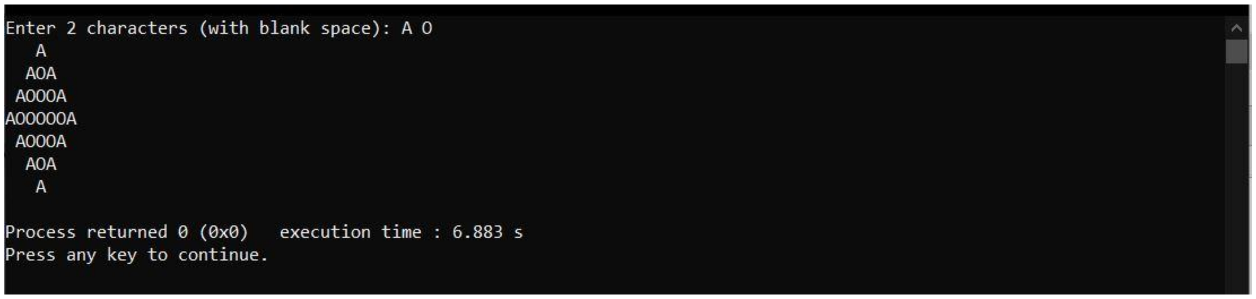
1. **Problem specification:**
   In this problem we have to display a diamond shape on console using to different characters.
2. **Analysis:**
   Diamond will print using 2 different characters entered by user. First input character will print ONLY in the boundary of diamond and second input character will print inside the diamond boundary.
3. **Design:**
   To solve this problem, we need to follow these steps:
   a. Program must have two char variables to store user inputs.
   b. Program should ask user to enter 2 different characters.
   c. Use those inputs to display a diamond shape on console.

4. **Implementation:**
   There are 3 different functions in this program
   a. Info function: This function will take two parameters as call by reference and store 2 different user inputs into those parameters.
   b. Diamond function: This function will take 2 parameters and display them in a diamond shape, where first character is boundary of diamond and second character will print inside the boundary.
   c. Main function: This function will drive the program. First it will declare two characters variables, character1 and charater2 to store use value. Then main function will call info function with call by reference to get user inputs and finally it will call diamond function with user inputs to be displayed.
5. **Testing and verification:**
   I have tested the program with different user inputs.
   1. Test case 1: With + and # inputs



```
Enter 2 characters (with blank space): + #
   +
  +#+
 +###+
+#####+
 +###+
  +#+
   +

Process returned 0 (0x0)   execution time : 11.044 s
Press any key to continue.
```

   2. Test case 2: With A and O inputs



```
Enter 2 characters (with blank space): A O
   A
  AOA
 AOOOA
AOOOOOA
 AOOOA
  AOA
   A

Process returned 0 (0x0)   execution time : 6.883 s
Press any key to continue.
```

Exercise: 2

Part I: Source code

```
#include <stdio.h>

#include <stdlib.h>


// function to calculate gross pay

float calculateGrossPay(int hours, float gbp){

    float grossPay;


    // if hours <= 38 then no over pay, else we will use 1.5 times the GBP

    if(hours <= 38){

        grossPay = hours * gbp;

    }else{

        grossPay = 38 * gbp;

        grossPay += (hours - 38) * (gbp * 1.5);

    }


    return grossPay;

}


float calculateTax(float grossPay){

    float tax = 0;


    // if gross pay <= 300 then simple 0.15 tax

    if(grossPay <= 300){

        tax = grossPay * 0.15;

    } else if(grossPay <= 500){

    // if gross pay <= 500, then 0.15 for first 300 and 0.2 for the rest

        grossPay -= 300;

        tax = 300 * 0.15;

        tax += grossPay * 0.2;

    } else if(grossPay <= 650){

    // if gross pay <= 650, then 0.15 for first 300 and 0.2 for next 200 and 0.25 from the rest

        grossPay -= 300;

        tax = 300 * 0.15;

        tax += 200 * 0.2;
        grossPay -= 200;

        tax += grossPay * 0.25;

    } else{

    // if gross pay is > 650, then 0.15 for first 300 and 0.2 for next 200 and 0.25 from the next
150 and 0.3 for the rest

        grossPay -= 300;

        tax = 300 * 0.15;

        tax += 200 * 0.2;

        grossPay -= 200;

        tax += 150 * 0.25;

        grossPay -= 150;

        tax += grossPay * 0.3;

    }


    return tax;

}


int main(){

    int choice, hours=0;

    float gbp=0, grossPay, tax;


    // pay rate menu

    printf("Enter the number corresponding to the desired pay rate: \n");

    printf("1-  8.50 GBP/hour      2-  9.90 GBP/hour \n");

    printf("3- 22.40 GBP/hour      4- 30.00 GBP/hour \n");


    // user input choice 1-4

    printf("\n: ");

    scanf("%d", &choice);


    // if choice is not from 1-4

    if(choice < 1 || choice > 4){

        printf("Wrong choice !! \n");

        exit(0);

    }


    printf("Enter number of hours worked in week: ");
    scanf("%d", &hours);

    // GBP according to user selected pay
    if(choice == 1){
        gbp = 8.5;
    } else if(choice == 2){
        gbp = 9.9;
    } else if(choice == 3){
        gbp = 22.4;
    } else if(choice == 4){
        gbp = 30;
    }

    grossPay = calculateGrossPay(hours, gbp);
    tax = calculateTax(grossPay);

    printf("\nGross Pay: %.2f \n", grossPay);
    printf("Taxes: %.2f \n", tax);
    printf("Net Pay: %.2f \n", grossPay - tax);

    return 0;
}
```

Part 2: Software development process

1 **Problem specification:**
In this problem, we have to calculate the gross pay, taxes and net pay of user using the selected pay rate and number of worked hours of user.

2 **Analysis:**
User will input an integer value to choose the desired pay rate from the menu and second input will be number of hours worked by user. Using these values program will calculate and display the gross pay, tax and net pay.

3 **Design:**
To solve this problem, we need to follow these steps:

a. Select pay rate from menu and calculate the gross pay.
b. Using that calculated gross pay, program will calculate the tax.
c. Finally, program will subtract the tax from gross pay to calculate the net pay.

4 Implementation:

There are 3 functions in the program:

a. CalculateGrossPay function:
   This function will calculate the gross pay if number of hours > 38 then program will increment GBP by 1.5 for over 38 hours.

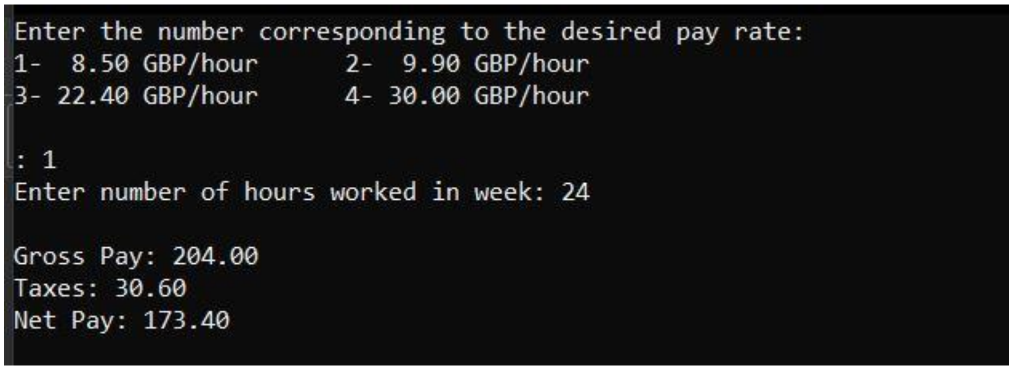
b. CalculateTax function:
   This function will calculate the tax apply to user using their gross pay, which will be pass by parameters to this function, tax formula will be

Tax rates:	15% of the first £300
	20% of the next £200
	25% of the next £150
	30% of the rest

c. Main function:
   This function will drive the whole program, it will print the menu from which user can select the pay rate. Then using user choice, it will specify the GBP value. Then it will call the calculateGrossPay with hours and GBP values as parameters. Then it will call the calculateTax function with grossPay as parameter. Finally, this function will print the gross pay, tax, and net pay of user.

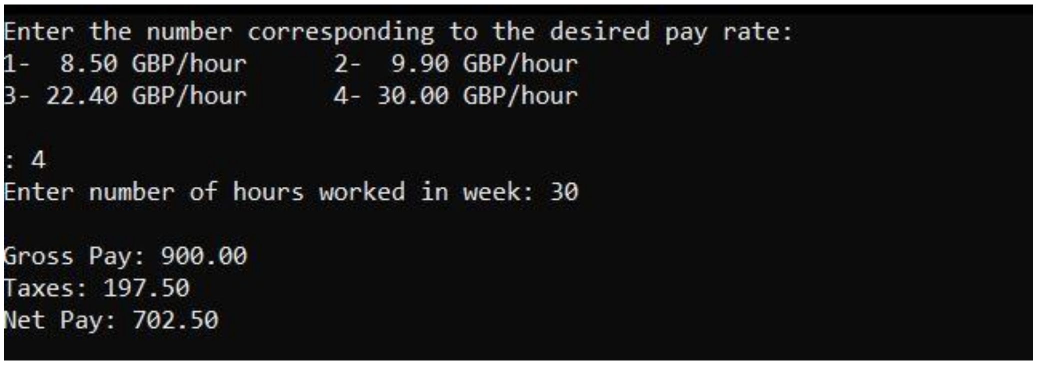
5 Testing and verification:

a. Test case 1: With 24 hours at 8.50 GBP/hour



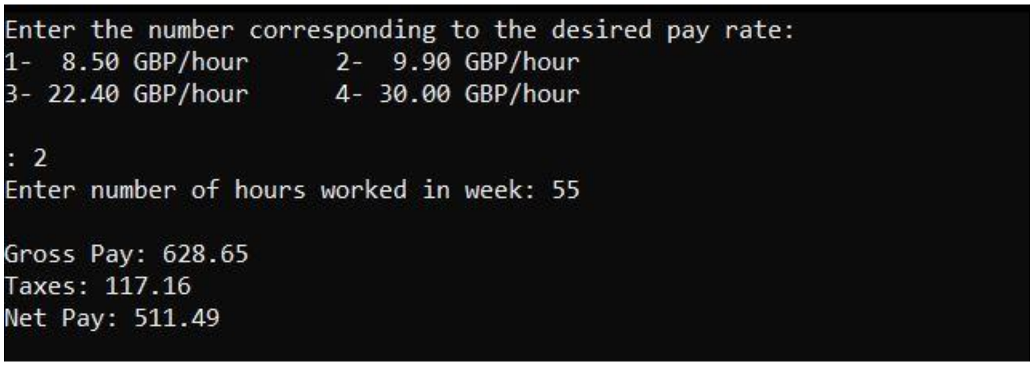
```
Enter the number corresponding to the desired pay rate:
1-  8.50 GBP/hour      2-  9.90 GBP/hour
3- 22.40 GBP/hour      4- 30.00 GBP/hour

: 1
Enter number of hours worked in week: 24

Gross Pay: 204.00
Taxes: 30.60
Net Pay: 173.40
```

b. Test case 2: With 30 hours at 30 GBP/hour



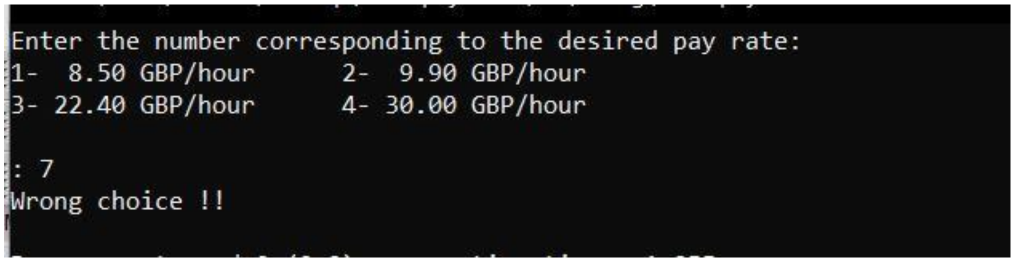
```
Enter the number corresponding to the desired pay rate:
1-  8.50 GBP/hour      2-  9.90 GBP/hour
3- 22.40 GBP/hour      4- 30.00 GBP/hour

: 4
Enter number of hours worked in week: 30

Gross Pay: 900.00
Taxes: 197.50
Net Pay: 702.50
```

c. Test case 3: With 55 hours at 9.90 GBP/hour



```
Enter the number corresponding to the desired pay rate:
1-  8.50 GBP/hour      2-  9.90 GBP/hour
3- 22.40 GBP/hour      4- 30.00 GBP/hour

: 2
Enter number of hours worked in week: 55

Gross Pay: 628.65
Taxes: 117.16
Net Pay: 511.49
```

d. Test case 4: With wrong menu choice



```
Enter the number corresponding to the desired pay rate:
1-  8.50 GBP/hour      2-  9.90 GBP/hour
3- 22.40 GBP/hour      4- 30.00 GBP/hour

: 7
Wrong choice !!
```