**Introduction to Programming in C**

**Assignment 1**

**Exercise 1**

**Part I**

#include <stdio.h>

/\*Functions used in code\*/

void info(char \*x, char \*y, char \*z);

void pyramid(char a, char b, char c);

/\*Main function\*/

void main (void)

{

char opt1, opt2, opt3; /\*Declares the chars that the users options are assigned to\*/

info(&opt1, &opt2, &opt3); /\*Calls the info function\*/

pyramid(opt1, opt2, opt3); /\*Calls the pyramid function\*/

}

/\*This function prints instructions to the user, gets the options from the user and then outputs the options\*/

void info(char \*x, char \*y, char \*z)

{

/\*Prints instruction to the user and gets the input\*/

char opt[2];

printf("Enter 3 characters: ");

scanf("%s", opt);

/\*This changes the value at the memory address. Pointers have been used so that multiple characters can be returned\*/

\*x = opt[0];

\*y = opt[1];

\*z = opt[2];

}

/\*Functions takes the characters as input and uses them to print pyramid\*/

void pyramid(char a, char b, char c)

{

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

printf("\n %c%c%c", a, b, a);

printf("\n %c%c%c%c%c", a, b, c, b, a);

printf("\n %c%c%c%c%c%c%c", a, b, c, c, c, b, a);

printf("\n %c%c%c%c%c%c%c%c%c", a, b, c, c, c, c, c, b, a);

printf("\n%c%c%c%c%c%c%c%c%c%c%c", a, b, c, c, c, c, c, c, c, b, a);

}

**Part II**

1. Specification:

The program is required to take three characters from the user and use those characters to create a pyramid.

1. Analysis:

Inputs: Characters.

Outputs: The characters in the shape of a triangle.

Other relevant aspects: Characters separated into three variables

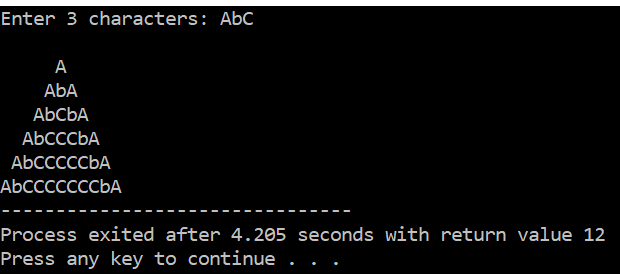
1. Algorithm:
2. Get the information from the user.
   1. Prints the instructions to the user.
   2. Get the three characters from the user.
   3. Return three characters to the main function by changing the value at their memory address.
3. Create the pyramid
   1. Take the three characters as input variables.
   2. Print each line of the pyramid separately
4. Implementation:

The code in part one features comments that explain how the code has been implemented and describes the functions.

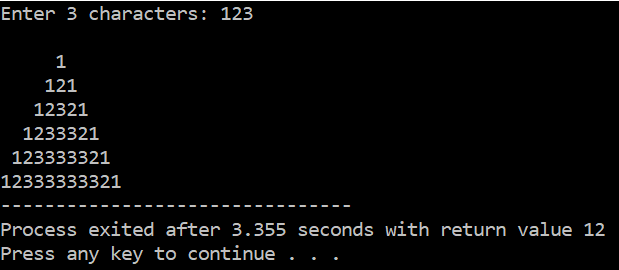
1. Testing and verification:

The program has been tested 4 times using different input

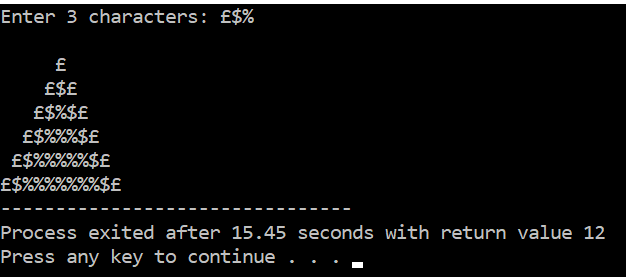
1. “AbC” – This test uses both uppercase and lower-case characters.



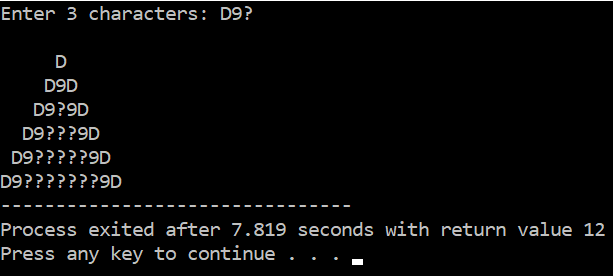
1. “123” – This test uses numbers.



1. “£$%” – This test uses symbols.



1. “D9?” – This test uses a letter, number, and symbol.



In all test cases that I have done the code appears to work as required and completes its objective.

**Exercise 2**

**Part I**

#include <stdio.h>

/\*Functions included in program\*/

void getinfo(char \*getvehicle, float \*getbuiltup, float \*getsingle, float \*getdual);

float fuel\_a(float builtupcalc, float singlecalc, float dualcalc);

float fuel\_b(float builtupcalc, float singlecalc, float dualcalc);

float calc\_salary(float total\_fuel\_cost);

void calc\_total(float total\_fuel\_cost, float get\_salary);

int main(void)

{

/\*Declares the variables\*/

char vehicle, a, b;

float builtup, single, dual, total\_fuel, salary;

/\*Step 1: Runs function to get the information from user\*/

getinfo(&vehicle, &builtup, &single, &dual);

/\*Step 2 (If option A selected): If statement that if true

runs function to calculate total cost of fuel\*/

if(vehicle == 'A')

{

total\_fuel = fuel\_a(builtup, single, dual);

}

/\*Step 2 (If option B chosen): If statement that if true

runs function to calculate total cost of fuel\*/

if(vehicle == 'B')

{

total\_fuel = fuel\_b(builtup, single, dual);

}

/\*Step 3: Runs function to calculate the salary\*/

salary = calc\_salary(total\_fuel);

/\*Step 4: Calculates the total cost overall\*/

calc\_total(total\_fuel, salary);

}

/\*This function is intended to get the input from the user and

then return them to the main function. Uses pointers so

that multiple values could be returned.\*/

void getinfo(char \*getvehicle, float \*getbuiltup, float \*getsingle, float \*getdual)

{

/\*Gets the vehicle type from the user and flushes so that future use of

scanf works correctly\*/

printf("Enter A or B car type: ");

scanf("%s", \*&getvehicle);

fflush(stdin);

/\*Makes program more robust. Converts incorrect case to uppercase\*/

if(\*getvehicle == 'a')

{

\*getvehicle = 'A';

}

if(\*getvehicle == 'b')

{

\*getvehicle = 'B';

}

/\*Gets distances from user in miles\*/

printf("Enter distance traveled in built up areas in miles: ");

scanf("%f", \*&getbuiltup);

fflush(stdin);

printf("Enter distance traveled in single cariageway: ");

scanf("%f", \*&getsingle);

fflush(stdin);

printf("Enter distance traveled in dual cariageway or motorway: ");

scanf("%f", \*&getdual);

fflush(stdin);

}

/\*Function to calculate total cost of fuel if vehicle A has been selected and

return result\*/

float fuel\_a(float builtupcalc, float singlecalc, float dualcalc)

{

float final;

/\*These floats contain how many miles per gallon for vehicle A.

They are also used later in the function\*/

float a = 40; /\*Shows how many miles per gallon in built up areas\*/

float b = 50; /\*Shows how many miles per gallon in single carriageway\*/

float c = 60; /\*Shows how many miles per gallon in built up dual carriageway\*/

float litre = 4.54609; /\*Conversion from mile to litre\*/

float price = 1.5; /\*Price in GBP for 1 litre\*/

/\*Calculates the final cost of the fuel\*/

builtupcalc = (builtupcalc \* (1/a));

singlecalc = (singlecalc \* (1/b));

dualcalc = (dualcalc \* (1/c));

final = builtupcalc + singlecalc +dualcalc;

final = final \* litre;

final = final \* price;

/\*Prints result. Return result to main function\*/

printf("The total cost of fuel for the journey is: %.2f GBP\n", final);

return final; // Return ends the function

}

/\*Function to calculate total cost of fuel if vehicle B has been selected

and return result\*/

float fuel\_b(float builtupcalc, float singlecalc, float dualcalc)

{

float final;

/\*These floats contain how many miles per gallon for vehicle B. They are also used later

in the function\*/

float a = 45; /\*Shows how many miles per gallon in built up areas\*/

float b = 55; /\*Shows how many miles per gallon in single carriageways\*/

float c = 65; /\*Shows how many miles per gallon in dual carriageways\*/

float litre = 4.54609; /\*Conversion from mile to litre\*/

float price = 1.3; /\*Price in GBP for 1 litre\*/

/\*Calculates the final cost of the fuel\*/

builtupcalc = (builtupcalc \* (1/a));

singlecalc = (singlecalc \* (1/b));

dualcalc = (dualcalc \* (1/c));

final = builtupcalc + singlecalc +dualcalc;

final = final \* litre;

final = final \* price;

/\*Prints result. Return result to main function\*/

printf("The total cost of fuel for the journey is: %.2f GBP\n", final);

return final; // Return ends the function

}

/\*Function to calculate the salary\*/

float calc\_salary(float total\_fuel\_cost)

{

float salary;

/\*If statements determines how much they would have earned from work

up to their current level of pay and then calcualtes everything after that\*/

if(total\_fuel\_cost >= 60)

{

total\_fuel\_cost = total\_fuel\_cost - 60;

salary = 140;

salary = salary + (5 \* total\_fuel\_cost);

total\_fuel\_cost = 0; /\*Prevents it from causeing following if statements to become true\*/

}

if(total\_fuel\_cost >= 30)

{

total\_fuel\_cost = total\_fuel\_cost - 30;

salary = 50;

salary = salary + (3 \* total\_fuel\_cost);

total\_fuel\_cost = 0; /\*Prevents it from causeing following if statements to become true\*/

}

if(total\_fuel\_cost >= 10)

{

total\_fuel\_cost = total\_fuel\_cost - 10;

salary = 10;

salary = salary + (2 \* total\_fuel\_cost);

total\_fuel\_cost = 0; /\*Prevents it from causeing following if statements to become true\*/

}

if(total\_fuel\_cost > 0)

{

salary = total\_fuel\_cost;

}

/\*Prints result to user and returns to main function\*/

printf("The total salary is: %.2f GBP\n", salary);

return salary;

}

/\*Function takes the total costs of fuel and salary and outputs total to user\*/

void calc\_total(float total\_fuel\_cost, float get\_salary)

{

float overall\_cost = total\_fuel\_cost + get\_salary;

printf("The total cost of the journey is: %.2f GBP", overall\_cost);

}

**Part II**

1. Specification:

This program calculates the cost of fuel, salary, and the total cost of both combined. The program bases this calculation on unleaded (vehicle A) fuel costing 1.50 GBP and diesel (vehicle B) fuel costing 1.30 GBP. The variables used are listed below.

Vehicle A:

* Uses unleaded fuel which costs 1.50 GBP per litre
* Fuel consumption in built up areas is 40 miles per gallon
* Fuel consumption in single carriageway is 50 miles per gallon
* Fuel consumption in dual carriageway or motorway is 60 miles per gallon

Vehicle B:

* Uses diesel fuel which costs 1.30 GBP per litre
* Fuel consumption is 45 miles per gallon
* Fuel consumption is 55 miles per gallon
* Fuel consumption is 65 miles per gallon

Other variables:

* The conversion that has been used it 1 gallon = 4.54609 litres
* Driver is paid £1 per £1 spent on fuel for first £10
* Driver is paid £2 per £1 spent on fuel for next £20
* Driver is paid £3 per £1 spent on fuel for first £30
* Driver is paid £5 per £1 spent on fuel for everything over £60

1. Analysis:

**Inputs:** Type of vehicle (A or B), miles travelled in built up area, miles travelled in single carriageway, miles travelled in dual carriageway or motorway

**Outputs:** Cost of fuel, cost of salary, and total cost of journey

**System Constraints:** Required to return more than one variable to main function but c programming language can only return one. Used pointer to fix this by changing the value stored in the memory address.

Have to use fflush(stdin) to clear the buffer so that scanf can be used correctly multiple times.

1. Algorithm:

* Step 1: Ask the user to input information
  + Vehicle type (A or B)
  + Checks to fix case of vehicle type if user has used “a” or “b”
  + Distance travelled in built up area in miles
  + Distance travelled in single carriageway in miles
  + Distance travelled in dual carriageway or motorway in miles
  + Returns values to main function
* Step 2: Calculate total fuel cost (Similar for both option A and B but they use some different variables to calculate)
  + Calculate how much fuel used in built up areas
  + Calculate how much fuel used in single carriage way
  + Calculate how much fuel used in dual carriageway or motorway
  + Add up all fuel usage
  + Calculate cost of all fuel used.
  + Print result to user
* Step 3: Calculate salary
  + If they have spent over £60 add what they would have earned up to that point and calculate remaining salary.
  + If they have spent over £30 add what they would have earned up to that point and calculate remaining salary.
  + If they have spent over £10 add what they would have earned up to that point and calculate remaining salary.
  + If they have spent less than £10 but more than £0 calculate salary.
  + Print result to user
* Step 4: Calculate total cost overall
  + Add cost of fuel and salary
  + Print the result to user

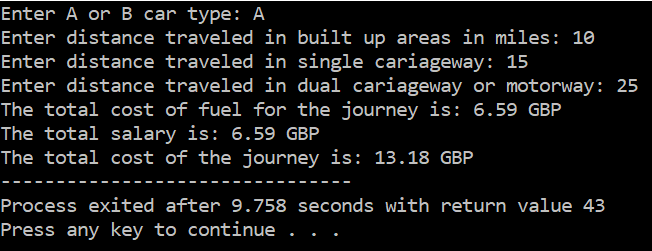
1. Implementation

The code in part one features comments that explain how the code has been implemented and describes the functions.

1. Testing and verification
   1. Case I

Inputs:

* Vehicle: A
* Miles travelled in built up areas = 10
* Miles travelled in single carriageway = 15
* Miles travelled in dual carriageway or motorway = 25



Results calculated mathematically:

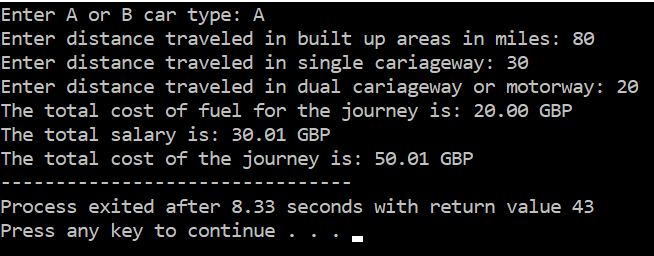
* Fuel cost = (((10/40) + (15/50) + (25/60)) \* 4.54609) \* 1.5 = 6.5918305 = 6.59 GBP
* Salary = 6.5918305 \* 1 = 6.5918305 = 6.59 GBP
* Total cost = 6.5918305 + 6.5918305 = 13.183661 = 13.18 GBP

Evaluation: The results produced by the program match what has been calculated mathematically so the program appears to work correctly.

* 1. Case II

Inputs:

* Vehicle: A
* Miles travelled in built up areas = 80
* Miles travelled in single carriageway = 30
* Miles travelled in dual carriageway or motorway = 20



Results calculated mathematically:

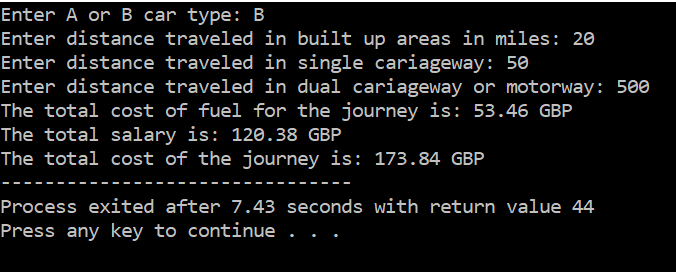
* Fuel cost = (((80/40) + (30/50) + (20/60)) \* 4.54609) \* 1.5 = 20.002796 = 20.00 GBP
* Salary = 10 + (10.002796 \* 2) = 30.005592 = 30. 01 GBP
* Total cost = 50.008388 = 50.01 GBP

Evaluation: The results produced by the program match what has been calculated mathematically so the program appears to work correctly.

* 1. Case III

Inputs:

* Vehicle: B
* Miles travelled in built up areas = 20
* Miles travelled in single carriageway = 50
* Miles travelled in dual carriageway or motorway = 500



Results calculated mathematically:

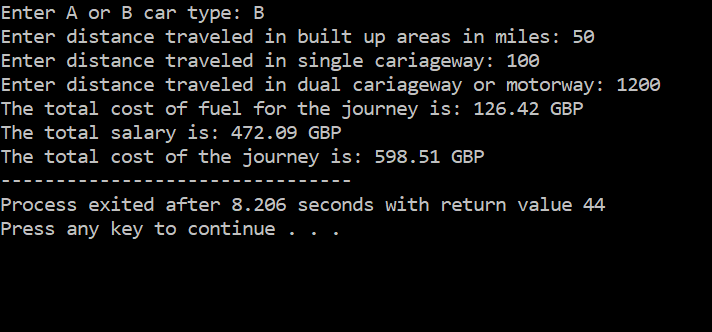
* Fuel cost = (((20/45) + (50/55) + (500/65)) \* 4.54609) \* 1.3 = 53.4601816 = 53.46 GBP
* Salary = 10 + 40 + (23.4601816 \* 3) = 120.3805448 = 120.38 GBP
* Total cost = 173.8407264 = 173.84 GBP

Evaluation: The results produced by the program match what has been calculated mathematically so the program appears to work correctly.

* 1. Case IV

Inputs:

* Vehicle: B
* Miles travelled in built up areas = 50
* Miles travelled in single carriageway = 100
* Miles travelled in dual carriageway or motorway = 1200



Results calculated mathematically:

* + - Fuel cost = (((50/45) + (100/55) + (1200/65)) \* 4.54609) \* 1.3 = 126.4180381 = 126.42 GBP
    - Salary = 10 + 40 + 90 + (66.4180381 \* 5) = 472.0901905 = 472.01 GBP
    - Total cost = 472.0901905 + 126.4180381 = 598.5082286 = 598.51 GBP

Evaluation: The results produced by the program match what has been calculated above mathematically so the program appears to work correctly.

Final evaluation: All test cases have provided the same result as was calculated mathematically so I believe that the program works correctly.