

C Programming

Learning Package 6

Applications 1

Dr Scott Turner

School of Science and Technology

University of Northampton



 **Introduction**

# In this Learning Package, you are going to solve some programming problems using techniques you have met in the first five learning packages.

**Aims**

At the end of this Learning Package, the reader should feel more confident in the ability to write programs using techniques they have met so far.

**Study guide**

Session 1:

Task 6.1

Session 2:

Task 6.2

Task 6.3

Session 3:

Task 6.4

Session 4:

Task 6.5

Task 6.1

Read in chapter 7 the sections from page 153 to page 165 of the module textbook.

Using the program below in place of program 7.4 on page 165, complete and execute this program.

#include <stdio.h>

#include<stdlib.h>

double process(); double getnum();

double evaluate(double x, char op, double y); char getnextchar();

int isadigit(char c);

int main()

{

printf("\nEnter your calculation: "); printf("%f",process());

return 0;

}

/\*in place of this comment put the definition for the following functions

evaluate() program 7.3

process() program 7.1

getnum() program 7.2

getnextchar() program 6.8 isadigit() program 6.9\*/

# Replace the section below with the appropriate functions

/\*in place of this comment put the definition for the following functions

evaluate() program 7.3

process() program 7.1

getnum() program 7.2

getnextchar() program 6.8 isadigit() program 6.9\*/

E.g. replace getnextchar() program 6.8

# With

char getnextchar()

{

char c; do {

c=getchar();

}while(c==' '); return(c);

}

# Task 6.2

1. Write a program that
	* includes a function to calculate the total resistance of four parallel resistors;
	* All entered values must be tested so that only resistance values greater than zero are acceptable.

1 1 

*RT R*1

# 1 1 1

*R*2 *R*3 *R*4

Equation 6.1

# Test data must be selected to check the program works.

1. Put in comments in your programs to explain the operation of the program.

Task 6.3

Using the program developed in Task 6.2 as a starting point, write a program that lets the user select from a menu the following options:

* + The total resistance of four parallel resistors
	+ The total resistance of four series resistors
	+ The total capacitance of four parallel capacitors
	+ The total capacitance of four series capacitors
	+ Exit the menu.

Task 6.4

1. Take program 4.1 on page 90-92 of the module textbook or task 3.1 in Learning Package 3.1 and modified the programs so these extra features are included:
	* Only three five-pound notes are given out in one transaction.
	* Change can include two-pound coins.

There are unlimited supplies of the others coins and notes. The program must still give the minimum number of coins.

1. Choose appropriate test data to prove that these old and new features all work.
2. Put in comments in your programs to explain the operation of the program.

Task 6.5

1. Write a program that will work out the potential difference between A and B in figure 6.1 for values of R1,R2,Rx and R3 entered by the user.

Hint: You might want to look at the Module TDL1001 Electronic Principles or a book on electronic principles for Wheatstone’s bridge for more detail on how to tackle this problem.

Figure 6.1

# Where the user enters

(i) R1=20 , R2=10,Rs=10, and Rx= 20.

(ii) R1=20 , R2=20,Rs=10, and Rx= 20.

1. Put in comments in your programs to explain the operation of the program.

**Selected tasks**

Task 6.2

#include <stdio.h>

#include <stdlib.h>

float p\_resistance(float R1, float R2, float R3, float R4); int main()

{

float R1,R2,R3,R4;

int flag=0; while (flag==0)

{

printf("\nEnter R1: "); scanf("%f",&R1); if(R1>0)

flag=1;

}

flag=0;

while (flag==0)

{

printf("\nEnter R2: "); scanf("%f",&R2); if(R2>0)

flag=1;

}

flag=0;

while (flag==0)

{

printf("\nEnter R3: "); scanf("%f",&R3); if(R3>0)

flag=1;

}

flag=0;

while (flag==0)

{

printf("\nEnter R4: "); scanf("%f",&R4); if(R4>0)

flag=1;

}

printf("\nTotal Resistance: %f\n",p\_resistance(R1, R2, R3, R4)); system("PAUSE");/\*left these two lines in, so when the program\*/ return 0; /\*is run the results can be seen\*/

}

float p\_resistance(float R1, float R2, float R3, float R4)

{

return(1/(1/R1+1/R2+1/R3+1/R4));

/\*functions do not need to be big\*/

}

# Task 6.4

#include <stdio.h>

#include <stdlib.h>

void main()

{

int five\_pound,pound,fifty,two\_pound,rest\_p, twenty,ten,five,two,one,pence,pounds,rest;

float money,price,change;

printf("\n Enter price of goods: "); scanf("%f",&price);

printf("\n Now enter the money tendered: "); scanf("%f",&money);

change =money-price; if (change<0){

printf("\n You haven't enough money"); exit(0);

}

printf("\n you need %f in change\n",change); pounds=change;

five\_pound=pounds/5; if (five\_pound<3)

{

rest\_p=pounds%5;

}

else

{

rest\_p=pounds%5+(five\_pound-3)\*5.00; five\_pound=3;

}

two\_pound=rest\_p/2; pound=rest\_p%2; pence=100\*(change +0.005-pounds); fifty=pence/50; rest=pence%50; twenty=rest/20; rest=rest%20;

ten=rest/10; rest=rest%10; five=rest/5; rest=rest%5; two=rest/2; one=rest%2;

printf("\n Change can be made up as follows: \n\n"); if (five\_pound>0) {

printf("\t %d five pound note",five\_pound); if (five\_pound>1)

printf("s");

printf("\n");

}

if (two\_pound>0) {

printf("\t %d two pound coin",two\_pound); if (two\_pound>1)

printf("s");

printf("\n");

}

if (pound>0) {

printf("\t %d pound coin",pound); if (pound>1)

printf("s");

printf("\n");

}

if (fifty>0) {

printf("\t %d fifty pence coin",fifty); if (fifty>1)

printf("s");

printf("\n");

}

if (twenty>0) {

printf("\t %d twenty pence coin",twenty);

if (twenty>1)

printf("s");

printf("\n");

}

if (ten>0) {

printf("\t %d ten pence coin",ten); if (ten>1)

printf("s");

printf("\n");

}

if (five>0)

printf("\t %d five pence coin",five); if (two>0) {

printf("\t %d two pence coin",two); if (two>1)

printf("s");

printf("\n");

}

if (one>0)

printf("\t %d one pence coin",one);

printf("\nThank you....\n"); system("PAUSE"); return(0);

}