2. (c) Consider the following C program: 2011/12

```
#include "stdio.h"
void main(void){
  char message[80]="I love C programming";
  int i=0, count=0;
  while (message[i]!='\0'){
    if ((message[i]=='c') || (message[i]=='C')){
      count++;
    }
    i++;
  }
  printf("\"%s\" contains %d c's\n", message, count);
}
```

function

fn prototype
call function

Re-write the above program so that all the code for determining the number of **c**'s in the string **message** is contained in a function called **c_counter()** that you should define, while the declaration and initialization of **message** and the output is still done from **main()**.

[10 Marks]

```
# include "stdio.h"
 void main (void) {
  char message [80] = "I love C programming";
                                 [ ]
  int c_counter ( char str[]);
  printf("\"%s\" contains %d c's\n", message,
            c - counter (message)) ;
     }

int C_counter ( char s[] )
 int i = 0 , count = 0 ;
 while ( s[i] != '\0'){
  if (( s[i] == 'c')||( s[i] == 'C')) {
    count ++ ;
    }
  i++ ;
}
```

```
  return count ;     }
```

number of hours, minutes, and seconds for a given time in seconds. For example, 3,661 seconds is equivalent to 1 hour, 1 minute, and 1 second. Use the following function prototype:

```
void convert(int time, int *phrs, int *pmins, int *psecs);
```

Remember to use good programming standards, define the problem statement, show the design of the program, add comments to the code, and follow good formatting practices to lay out the code. Your answer should include the algorithm design and the code.

[20 Marks]

1. Problem statement.
Take in a integer (postive) number of seconds to convert to hours, min & sec

2. Main program, initatise variables, call program, print out results

3. function do conversion — number/3600 Hours
remainder *60 minutes
remainder seconds

```
void convert(int time, int *phrs, int *pmins, int *psecs) {
    *phrs = time/(60*60);

    time -= *phs*60*60    /* How much is left */

    *pmins = time/60;

    time -= *pmins*60;

    *psec = time;

}
```

```
#include<stdio.h>
void main(void) {
void convert(int tim ....);
int time, hrs, mins, secs;
printf("Enter time to convert in
        seconds: ");
scanf("%d", &time);
convert(time, &hrs, &mins, &secs);
```

[10 Marks]

(b) Consider the following C program:

```
#include <stdio.h>
/* DEFINITION OF FUNCTION "zerofinder" GOES HERE */
void main(void) {
 int i, array2[8]={1,-1,-1,0,1,0,-1,1};
 i = zerofinder(array2,8);
 if (i == -1) printf("\nno zero value in array2\n");
else printf("\nfirst 0 element array2 has index %d\n", i);
}
```

Write down the definition of function **zerofinder()** which returns the index of the first 0 value, or -1 if no 0 value is found, so that the output of the above program is:

```
first 0 element array2 has index 3
```

[15 Marks]

```
zerofinder( int A[], int size) {
j=0;
(j=0; j <size, j++) {
if    A[j] == 0 {
        return j;
        }}
return -1;
```

INT

4357.~~3810~~

(j) What is the screen output of the following fragment of C code?

```
float number = 435.73810, new ;
new = ((int)(10*number))/10.0 ;
printf("number is %7.2f\n",new);
```

435.70

number is __435.70

(k) What is the screen output of the following fragment of C code?

```
int i,j=2;
for (i=6;i>2;i--){
  j+=i;
  printf("j is %d\n",j);
}
```

j = j + i ;

j is 8
j is 13
j is 17
j is 20

(l) What is the screen output of the following fragment of C code?

```
double x=3.142;
printf("value is %4.2f",3*x);
```

9.~~4~~26

value is 9.43

(m) What is the screen output of the following fragment of C code?

```
int i=10;
while (i>5){
  i -= 2;
  printf("i is %d\n",i);
}
```

i = i - 2

i is 8
i is 6
i is 4

(n) What is the screen output of the following fragment of C code?

```
#include <stdio.h>
int f1(int a, int b, int c){
        return (b);
}
int f2(int a, int b){
        int c = f1(a, b, 3);
        return c+1;
}
void main()
{
        printf("result is %d\n", f1(-1,f2(2,4),2));
}
```

picks middle

add 1 to second integer

5

5

result is 5

(o) What is the screen output of the following fragment of C code?

```
int x = -2;
int y = 3;
int* p = &y;
*p = (*p)*x - (*p)*y;
printf("x is %d and y is %d\n", x, y);
```

*p → y

3 × -2   3 × 3   -6 - 9   -15 = y

x is -2 and y is -15

(r) What is the screen output of the following fragment of C code? The initial values in datafile.txt are:

0.2 80 16.0 Avg ~~[illegible]~~ 58.0
0.7 60 ~~[illegible]~~42.0
0.1 50 5.0 ~~[illegible]~~ 63.0

```
float prob, avg=0.0;
int quantity, num_values=0;
FILE *fptr;
fptr=fopen("datafile.txt", "r");
while(fscanf(fptr, " %f %d", &prob, &quantity)==2){
/* if return value from this fscanf() is not 2, */
/* the end-of-file indicator has been reached */
avg = avg + (prob*quantity);
}
fclose(fptr);
printf("average value is %.2f\n", avg);
```

average value is 63.00

(s) What are the contents of datafile.txt after the execution of this fragment of code? The initial values in datafile.txt are:

200
-54

-54
200

```
int var1, var2;
FILE *fp;
FILE *fptr;
fp=fopen("datafile.txt", "r");
fscanf(fp, "%d %d", &var1, &var2);
fclose(fp);
fptr=fopen("datafile.txt", "w");
fprintf(fptr, "%d\n%d", var2, var1);
fclose(fptr);
```

over-write

-54 200

(t) What is the screen output of the following fragment of C code?

```
double x=3.14159;
printf("value is %.3e",x);
```

Value is 3.142e+00