

UNIVERSITY OF PORTSMOUTH

FACULTY OF TECHNOLOGY

Department of Electronic and Computer Engineering

B142L – Introduction to Computing

U13746

Date: 29 May 2009

Time: 9:00 – 10:30

INSTRUCTIONS

Write your student ID number clearly on page 2.

Write your answers to all 10 questions within the spaces provided in this examination paper.

Handwritten notes are permitted with this examination.

Calculators permitted are:

Casio FX 85WA

Casio FX 83WA

Casio FX 85MS

Examiner:

Professor Ron Pethig, Chi Nguyen

Student ID Number

SOLUTIONS

QUESTION 1

a) Place an "X" in the box next to **3 terms** that are **basic data types** in the C language.

[3 Marks]

	constant
1	character
	variable
1	integer
	byte

	null
	time
1	float
	array
	string

b) Write a C program to ask and store in memory the name, age and exam mark for a student. Show or use all the terms selected in part (a).

[7 Marks]

Similar to:

```
int main( void ) [1]
{
    /* array of characters */
    char name[ 64 ] = { '\0' }; [1]
    int age = 0; [1]
    float mark = 0.0; [1]

    scanf( "%s", name ); [1]
    scanf( "%d", &age ); [1]
    scanf( "%f", &mark ); [1]
}
```

QUESTION 2

Modify the following source code to correct all errors. The program is intended to **ask for an input string, count the instances of character 'e' found in the string and display to output the total number found.** Line numbers have been provided for reference. Indicate specific line numbers in your answers when necessary. [10 Marks]

```
01: #include <string.h>
02:
03: int main( void )
04: {
05:     char text[ 512 ] = ( '\0' );
06:     int found = 10;
07:     int counter;
08:
09:     scanf( "%d", text );
10:
11:     for ( counter = 0; counter < strlen( text ); counter )
12:     {
13:         if ( text[ counter ] == 'e' )
14:             found;
15:     }
16:
17:     printf( "Found %d matching characters.", counter );
18: }
```

```
#include <string.h>

int main( void )
{
    char text[ 512 ] = { '\0' };           [2]
    int found = 0;                         [1]
    int counter;

    scanf( "%s", text );                   [2]

    for ( counter = 0; counter < strlen( text ); counter++ )   [2]
    {
        if ( text[ counter ] == 'e' )
            found++;                       [2]
    }

    printf( "Found %d matching characters.", found );           [1]
}
```

QUESTION 3

Consider the following source code. Line numbers have been provided for reference. Indicate specific line numbers in your answers when necessary.

```
01: int main( void )
02: {
03:     double temperature = 0.0; /* Celsius */
04: }
```

a) Modify the source code to **accept an input temperature from the keyboard.** [2 Marks]

Add between lines 3 and 4:

```
scanf( "%lf", &temperature ); [2]
```

b) Modify the source code to **display one of the following messages based on the value of the input temperature:** [8 Marks]

25 or above	"Hot day"
Less than 25 and 12 or above	"Warm day"
Less than 12	"Cold day"

Add between lines 3 and 4:

```
scanf( "%lf", &temperature );

if ( temperature >= 25.0 ) [3]
    printf( "Hot day" );
else if ( temperature >= 12.0 ) [3]
    printf( "Warm day" );
else [2]
    printf( "Cold day" );
```

QUESTION 4

- a) Place an "X" in the box next to **3 terms** that are most directly related to the use of **arrays** in a C program. [3 Marks]

<input type="checkbox"/>	int	<input checked="" type="checkbox"/>	element
<input checked="" type="checkbox"/>	type	<input checked="" type="checkbox"/>	length
<input type="checkbox"/>	shell	<input type="checkbox"/>	static
<input type="checkbox"/>	table	<input type="checkbox"/>	limit
<input type="checkbox"/>	range	<input type="checkbox"/>	shift

- b) Write a C program to accept as **input the age of 40 people**, stores the values in an array and **displays the youngest age** at the end. Show or use all the terms selected in part (a). [7 Marks]

Similar to:

```
int main( void )
{
    /* array type is integer and the array length is 40 */
    int age[ 40 ] = { 0 };
    int youngest = -1;
    int counter;

    for ( counter = 0; counter < 40; counter++ )
    {
        /* Save the input age into the current array element */
        scanf( "%d", &age[ counter ] );
        if ( ( age[ counter ] < youngest ) || ( youngest < 0 ) )
            youngest = age[ counter ];
    }

    printf( "The youngest age is %d", youngest );
}
```

QUESTION 5

Consider the following source code. Line numbers have been provided for reference. Indicate specific line numbers in your answers when necessary.

```
01: int main( void )
02: {
03:     int input;
04:
05:     scanf( "%d", &input );
06:
07:     do
08:     {
09:         if ( ( input % 4 ) == 0 )
10:             printf( "Blue" );
11:         else
12:             printf( "Green" );
13:
14:         if ( input > 10 )
15:             input = 10;
16:         else
17:             input--;
18:     } while ( input > 0 );
19: }
```

a) Describe all valid input values that would cause the following output text to be displayed:

BlueGreenGreenGreenBlueGreenGreenGreen

[3 Marks]

Integer value 8

[3]

b) Describe all valid input values that would cause the longest output text to be displayed.

[7 Marks]

The longest output text is:

GreenGreenGreenBlueGreenGreenGreenBlueGreenGreenGreen

[2]

Valid input values that would cause the output text above are:

Integer values equal to or higher than 11,

[3]

which are not multiples of 4

[2]

QUESTION 6

Consider the following source code. Line numbers have been provided for reference. Indicate specific line numbers in your answers when necessary.

```
01: #include <string.h>
02:
03: int main( void )
04: {
05:     char password[ 512 ] = { '\0' };
06:     int index;
07:
08:     scanf( "%s", password );
09: }
```

- a) Modify the source code and **replace the first and every other character of the input password with the character '*' (asterisk). Display the modified password.** [7 Marks]

Add between lines 8 and 9:

```
for ( index = 0; index < strlen( password ); index = index + 2 )           [4]
    password[ index ] = '*';                                             [2]

printf( "%s\n", password );                                             [1]
```

- b) Modify the source code to **display a warning message if the length of the input password is less than 4 characters.** [3 Marks]

Add between lines 8 and 9:

```
if ( strlen( password ) < 4 )                                           [2]
    printf( "The password is too short." );                             [1]
```

QUESTION 7

- a) Place an "X" in the box next to **3 terms** that are most directly related to the use of **selection and decision** in a C program. [3 Marks]

<input type="checkbox"/>	sprintf	<input type="checkbox"/>	other
<input type="checkbox"/>	sscanf	<input checked="" type="checkbox"/>	and
<input type="checkbox"/>	printf	<input type="checkbox"/>	to
<input type="checkbox"/>	scanf	<input checked="" type="checkbox"/>	or
<input type="checkbox"/>	for	<input checked="" type="checkbox"/>	if

- b) Write a C program to **ask for the user age and discount code**. The program should **display to output the appropriate ticket price** using the following information. Show or use all the terms selected in part (a). [7 Marks]

£5 Age \geq 65 and discount code = 10
£5 Age \leq 12 and discount code = 10
£8 Any other age and discount code

Similar to:

```
int main( void )
{
    int age, discount;

    scanf( "%d", &age );
    scanf( "%d", &discount );

    if ( ( ( age >= 65 ) || ( age <= 12 ) ) && ( discount == 10 ) )
        printf( "Ticket price is £5" );
    else
        printf( "Ticket price is £8" );
}
```

QUESTION 8

Modify the following source code to correct all errors. The program is intended to ask for a password until the input matches the "secret" text. Line numbers have been provided for reference. Indicate specific line numbers in your answers when necessary. [10 Marks]

```
01: #include <string.h>
02:
03: int compare( void );
04:
05: int main( void )
06: {
07:     char text[ 512 ] = [ '\0' ];
08:
09:     do
10:     {
11:         scanf( "%s", text );
12:     } while ( compare( password ) == 0 )
13: }
14:
15: int compare( char *password )
16: {
17:     if ( strcmp( "secret" ) == 0 )
18:         return 1;
19:     else
20:         return 0;
21: }
```

```
#include <string.h>

int compare( char* ); [2]

int main( void )
{
    char text[ 512 ] = { '\0' }; [2]

    do
    {
        scanf( "%s", text );
    } while ( compare( text ) == 0 ); [4]
}

int compare( char *password )
{
    if ( strcmp( password, "secret" ) == 0 ) [2]
        return 1;
    else
        return 0;
}
```

QUESTION 9

Write a C program that **accepts 100 input values representing the year of birth** for students currently in university. The **input values must be stored in an array**. **Input values less than 1980 or above 1995 must be replaced with a -1 value.** [10 Marks]

Similar to:

```
int main( void )
{
    int years[ 100 ];
    int index, input;

    for ( index = 0; index < 100; index++ )
    {
        scanf( "%d", &input );
        if ( ( input >= 1980 ) && ( input <= 1995 ) )
            years[ index ] = input;
        else
            years[ index ] = -1;
    }
}
```

QUESTION 10

a) Place an "X" in the box next to **3 terms** that are most directly related to the use of **iterative loops** in a C program. **[3 Marks]**

	default
	format
	global
1	while
1	body

	main condition
	error condition
1	test condition
	prototype
	template

b) Write a C program that **keeps asking for an input text string as long as the length of the input is greater than 10**. Show or use all the terms selected in part (a). **[7 Marks]**

```
Similar to:

#include <string.h>

int main( void )
{
    char text[ 512 ] = { '\0' }; [1]

    do
    {
        scanf( "%s", text ); /* body of the loop */ [2]
    } while ( strlen( text ) > 10 ); /* test condition */ [4]
}
```