

Before you begin your work, please create a new file folder on your computer. The name of the folder should be YourLastName\_YourFirstName. For example, if your name is John Smith your folder should be named Smith\_John. Please store all your C files in that folder. Name your C files simply 1.c, 2.c, 3.c, 4.c, 5a.c, and 5b.c. At the end of the exam you will have to copy that folder onto a USB memory stick. Also, you will have to upload your solutions to WebCT (just like a homework but this one is named "Midterm1 Lab").

Please DO NOT leave the room until you have copied your files onto the USB memory stick and also uploaded them on WebCT!!!

### 1. Pixel Color Conversion (RGB to YUV) (10 points)

The pixels on your monitor are typically represented in RGB color space. The letters stand for Red, Green, and Blue. The colors are encoded with three numbers, one for each color. The most popular format uses three integers in the range from 0 to 255 (8-bit color). Here are some examples of popular colors and their RGB representations:

R	G	B	// Name
0	0	0	// black
255	0	0	// red
0	255	0	// green
0	0	255	// blue
0	0	128	// navy blue
255	255	0	// yellow
165	42	42	// brown
255	255	255	// white

An alternative representation that is sometimes useful for computer vision applications is YUV. Here is how to convert from RGB to YUV:

$$Y = 0.299R + 0.587G + 0.114B$$
$$U = 0.492 (B - Y)$$
$$V = 0.877 (R - Y)$$

Write a complete C program that reads in 3 numbers representing the three colors (R, G, and B) and outputs the corresponding YUV colors.

### 2. miles/gallon to liters/100 kilometers (15 points)

Write a complete C program that converts the fuel consumption of a vehicle from miles per gallon to liters per 100 kilometers.

The program must do the following:

- \* ask the user to enter a number representing the miles/gallon
- \* calculate the conversion and print the result on the screen

Hint: 1 kilometer = 0.621 miles  
1 gallon = 3.785 liters

SAMPLE RUN:

This program converts miles/gallon to liters/100 kilometers  
Enter a value (miles/gallon): 50

50.000000 miles/gallon = 4.700970 liters/100 kilometers

### 3. Complex Numbers (15 points)

Write a complete C program that asks the user to enter the real and the imaginary parts of two complex numbers. The program must then calculate and print on the screen the result of adding, subtracting, multiplying, and dividing the two complex numbers. Each of the three operations must be performed by a separate function. The result of each operation must be printed by each function as we don't yet know how to make a function return two values (e.g., the four functions can call a separate printComplex function to help them print the number).

Math refresher:

Given two complex numbers  $Z1 = a + bi$  and  $Z2 = c + di$ , the four operations are defined as:

Addition:  $(a + bi) + (c + di) = (a+c) + (b+d)i$   
Subtraction:  $(a + bi) - (c + di) = (a-c) + (b-d)i$   
Multiplication:  $(a + bi) * (c + di) = (ac - bd) + (bc + ad)i$   
  
Division:  $\frac{(a + bi)}{(c + di)} = \frac{(ac + bd)}{(c*c + d*d)} + \frac{(bc - ad)}{(c*c + d*d)} i$

SAMPLE RUN:

=====

```
Z1 - Enter real part (a):      1
Z1 - Enter imaginary part (b): 2

Z2 - Enter real part (c) :     3
Z2 - Enter imaginary part (d): 4

Z1 = ( 1.0000) + ( 2.0000)i
Z2 = ( 3.0000) + ( 4.0000)i

Addition:      ( 4.0000) + ( 6.0000)i
Subtraction:   (-2.0000) + (-2.0000)i
Multiplication: (-5.0000) + (10.0000)i
Division:      ( 0.4400) + ( 0.0800)i
```

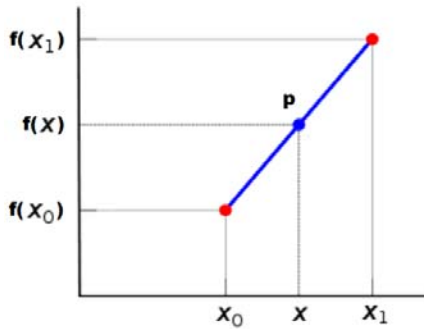
### 4. Loan Calculator (15 points)

Write a C program that calculates the monthly payments for a mortgage loan. The program must perform the following steps: 1) read the loan amount, the number of years, and the annual interest rate (APR) from the keyboard; 2) calculate the monthly interest rate from the annual interest rate (divide by twelve); 3) compute and print the monthly payment using the formula given below; 4) compute and print the total payment, which is equal to the monthly payment times 12 times the number of years.

$$\text{monthlyPayment} = \frac{\text{loanAmount} \times \text{monthlyInterestRate}}{1 - \frac{1}{(1 + \text{monthlyInterestRate})^{\text{numberOfYears} \times 12}}}$$

## 5. Linear Interpolation (15 points)

Linear interpolation is a common method for estimating the value of a function at a point  $x$  for which no data is available, but data exists for points located on either side of  $x$  on the horizontal axis. Given two points  $x_0$  and  $x_1$  and the value of the function evaluated at these points  $f(x_0)$  and  $f(x_1)$  using linear interpolation one can estimate the value of the function  $f(x)$  at another point  $x$  that lies between  $x_0$  and  $x_1$ . The interpolation formula is as follows:



$$f(x) = \frac{(x - x_0)}{(x_1 - x_0)} f(x_1) + \frac{(x_1 - x)}{(x_1 - x_0)} f(x_0)$$

In other words, the estimation is performed by fitting a straight line between the points  $(x_0, f(x_0))$  and  $(x_1, f(x_1))$  and using this line to estimate the values of the function in the interval  $(x_0, x_1)$ . Note that we don't even need to know the analytical form of the function; we only need to know its values at two points.

Write a C function that: 1) reads five numbers ( $x_0$ ,  $f(x_0)$ ,  $x_1$ ,  $f(x_1)$  and  $x$ ) from the keyboard; 2) calls a function that finds the value of  $f(x)$  at  $x$ ; 3) prints the interpolated value on the screen.

SAMPLE RUN #1:

```
Please enter x0: 1
Please enter f(x0): 1
Please enter x1: 3
Please enter f(x1): 3
```

```
Please enter x: 2
```

The interpolated value of the function at  $x=2$  is equal to 2

SAMPLE RUN #2:

```
Please enter x0: -1
Please enter f(x0): -2
Please enter x1: 2
Please enter f(x1): 4
```

```
Please enter x: 1.5
```

The interpolated value of the function at  $x=1.5$  is equal to 3