Engineering 12 - Spring, 1999

1. (18 points) A portion of a C program is given below. Fill in the missing code to calculate and display a table of n vs n^3, as shown below:

1	1
2	8
3	27
4	64
5	125
6	216
7	343
8	512
9	729
10	1000

```
/*   This program will calculate and display n cubed   */

#include <stdio.h>
#include <math.h>

main()
{
     int n;
     float n_cubed;



}  /* end of program */
```

2. (18 points) This question involves C fundamentals. It has three parts:

(a) Write a declaration for the following variables. Assign initial values as indicated.

(i) x is a double-precision variable whose initial value is 0.006

(ii) *ymax* is a floating-point variable whose initial value is 1.75×10^4 (use scientific notation)

(iii) *n* is an integer variable whose initial value is –8

(iv) *color* is a string whose initial value is *yellow*.

(b) Suppose *a* and *b* are integer variables. Write the necessary *scanf* and *printf* statements to prompt the user for input values of these two variables and then display their sum. Label the output accordingly.

(c) Suppose i and j are integer variables, and the value assigned to j will depend on the value of i, as indicated below:

If $0 \le i < 10$, then j = 5. Otherwise, j = 7.

Write the necessary C statement(s) to carry out this assignment.

3. (20 points) Match each of the program fragments shown below with the corresponding flowchart on the next page. Also determine the output of each program fragment.

1.
```
float s = 0;
int n = 0;
while (n < 5)
   n = n + 1;
s = s + 1.0/n;
printf("s = %f \n", s);
```

flowchart (a, b, c or d): _____ output: _____

2.
```
float s = 0;
int n = 0;
if (n < 5)
{
    n = n + 1;
s = s + 1.0/n;
}
printf("s = %f \n", s);
```

flowchart (a, b, c or d): _____ output: _____

3.
```
float s = 0;
int n = 0;
while (n < 5)
{
    n = n + 1;
s = s + 1.0/n;
}
printf("s = %f \n", s);
```

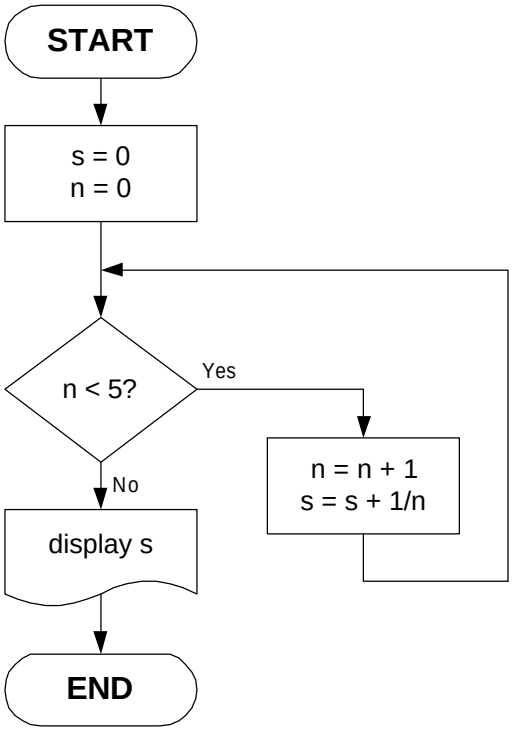
flowchart (a, b, c or d): _____ output: _____

4.
```
float s = 0;
int n = 0;
if (n < 5)
    n = n + 1;
else
s = s + 1.0/n;
printf("s = %f \n", s);
```

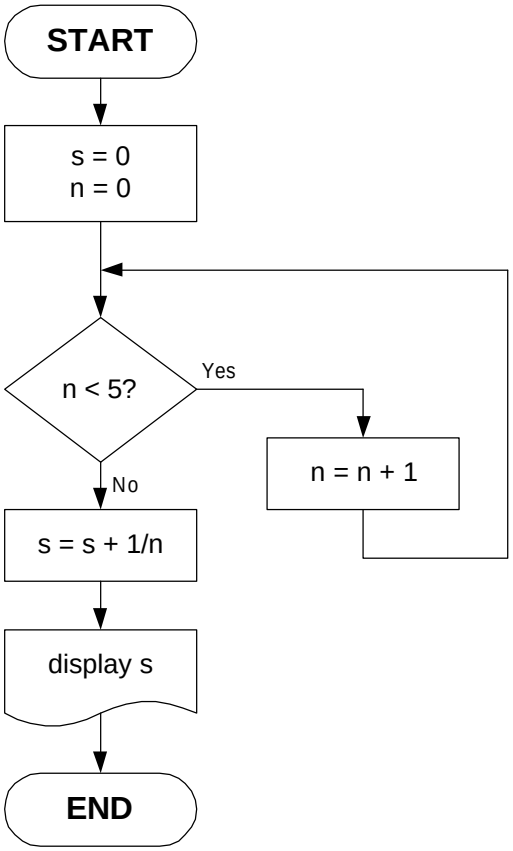
flowchart (a, b, c or d): _____ output: _____

(*flowcharts appear on the next page*)

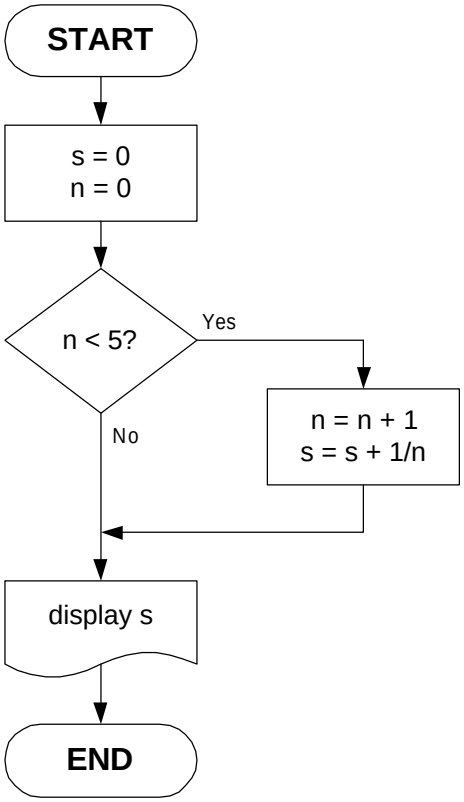
a)


START
s = 0
n = 0
n < 5?
Yes
No
n = n + 1
s = s + 1/n
display s
END

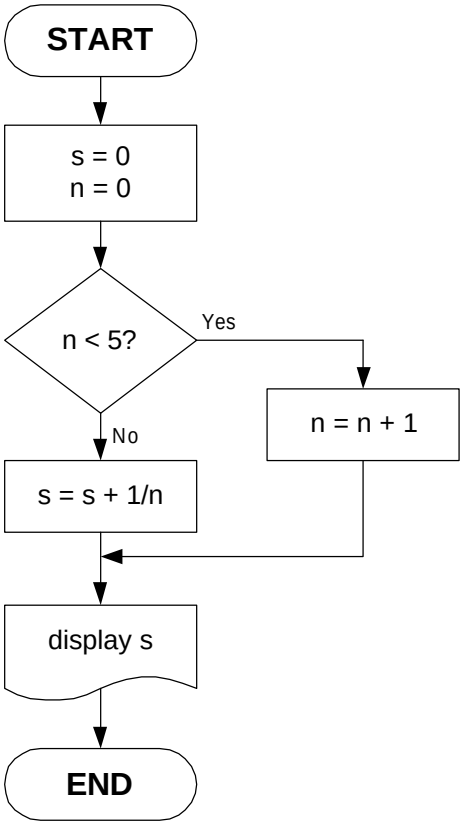
b)


START
s = 0
n = 0
n < 5?
Yes
No
n = n + 1
s = s + 1/n
display s
END

c)


START
s = 0
n = 0
n < 5?
Yes
No
n = n + 1
s = s + 1/n
display s
END

d)


START
s = 0
n = 0
n < 5?
Yes
No
n = n + 1
s = s + 1/n
display s
END

4. (20 points) This question pertains to the program shown below.

```
#include <stdio.h>

main()
{
     char id;
     int age, zones;
     float fare;

     printf("Please enter your age.");
     scanf("%d", &age);

     printf("How many PAT zones will you cross? (1,2,3)");
     scanf("%d", &zones);

     printf("Do you have a Pitt ID? (y/n)");
     scanf(" %c", &id);

     if ((id != 'y') && (age >= 6)) {

          if (age > 11)  {
               if (zones == 1)
                    fare = 1.25;
               else if (zones == 2)
                    fare = 1.6;
               else
                    fare = 1.95;
          }

          else
               if (zones == 1)
                    fare = 0.6;
               else if (zones == 2)
                    fare = 0.8;
               else
                    fare = 0.95;
     }

     else
          fare = 0;

     printf("Your fare for this ride is $ %.2f \n", fare);
}
```

(*Question continues on next page*)

Circle the correct answer to each of the following questions.

(a) What fare would an 11-year-old (without a Pitt ID) pay for a three-zone ride?

$ 0.00

$ 0.60

$ 0.95

$ 1.25

$ 1.95

None of the above

(b) What fare would a 20-year-old Pitt student pay for a two-zone ride?

$ 0.00

$ 0.60

$ 0.95

$ 1.25

$ 1.95

None of the above

(c) Which rider would pay a fare of $ 1.60 ?

Pitt Student	age 18	one-zone ride
CMU Student	age 18	one-zone ride
Child	age 9	two-zone ride
Child	age 9	three-zone ride

None of the above

(d) Which *group* of riders would pay a *total* fare of $ 1.60 ?

Adult age 38	and child age 5	w/o Pitt ID	one-zone ride
Adult age 38	and child age 5	with Pitt ID	two-zone ride
Adult age 38	and children ages 6 and 8	w/o Pitt ID	one-zone ride
Adult age 38	and children ages 6 and 8	with Pitt ID	two-zone ride

None of the above

5. (24 points) This question applies to the program shown on the following page. The lines are numbered for your convenience in responding to the question.

a) The program ran once but when the user attempted to run a second case the computer displayed the message: General Protection Exception. Explain why this occurred and state what is required to correct it.

b) When the error in part a) was corrected, the user entered values of 1, 2, and 3 for *a*, *b*, and *c* respectively and the program gave the correct result. However, when values of 2, 4, and 6 were entered (which should have given the same result), an incorrect result was obtained. Explain why this occurred and state what is required to correct it.

c) Explain why $a = 0$ must be treated as a special case while $b = 0$ and $c = 0$ do not.

d) If the errors in parts *a*) and *b*) are corrected and the user wishes to solve the equation

$$3x^2 - 5x + 6 = 0$$

which branch (2 real roots, 2 complex roots, or single repeated root) of the program is executed ?

<u>*Reminder*</u>: $x = \dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

```
/* FINDING THE ROOTS OF THE QUADRATIC EQUATION: ax^2 + bx + c = 0 */

#include<stdio.h>
#include<math.h>

main()
{

      float a, b, c, dscrmnt, u, v, x1, x2;
      char ans = 'y';

      while (ans == 'y') {               /* repeat while loop */

            printf("\nPlease enter the x^2 coefficient (a): ");
            scanf("%f", &a);
            printf("\nPlease enter the x coefficient( b): ");
            scanf("%f", &b);
            printf("\nPlease enter the constant( c): ");
            scanf("%f", &c);


            if( a == 0 )                 /* special case for a = 0 */

                  printf("\nThe single x value = %.4f", -c/b);

            else  {    /* determine values of discriminant and solution factors */

                  dscrmnt = b*b - 4*a*c;

                  /* calculate roots based on sign of the discriminant */

                  if( dscrmnt > 0) {              /* 2 distinct real roots */
                        x1 = (-b + pow(dscrmnt,0.5))/2*a;
                        printf("\nThe first root = %.4f", x1);
                        x2 = (-b - pow(dscrmnt,0.5))/2*a;
                        printf("\n\nThe second root = %.4f", x2);
                  }

                  else if (dscrmnt < 0) {         /* 2 distinct complex roots */
                        u = -b/2*a;
                        v = pow(-dscrmnt,0.5)/2*a;
                        printf("\nThe 1st complex root = %.4f + %.4fi", u, v);
                        printf("\n\nThe 2nd complex root = %.4f - %.4fi", u, v);
                  }

                  else                            /* single repeated root */
                        printf("\nThe repeated root = %.4f", -b/2*a);
            }
                  printf("\n\nAgain? (y/n) ");
                  scanf(" %c", ans);

      }     /* end of while loop  */
}
```

Note: `pow(x,0.5)` is equivalent to `sqrt(x)`