C Programming

Charudatt Kadolkar

IIT Guwahati

Grading Scheme

Written Examinations		70 %
Quizzes(2)	20%	
Midsem	30%	
Endsem	50%	
Laboratory Work		30 %

Textbooks

1. *The C Programming Language* by **Kernighan** and **Ritchie**, PHI.
2. *Computer Programming in C* by **V Rajaraman**, PHI.

Introduction

- Programming

  Set of Instructions, Manual, Recipies, ...

- Language

  C, Fortran, Pascal, Java, ...

How do we go about this?

Programs

Peas Pulao

Ingredients

3 onions

4 cloves

2ßtick cinnamon

2 bay leaves

4 tsps oil

1 cup basmati rice

1/4 kg. peas

2 cups hot water

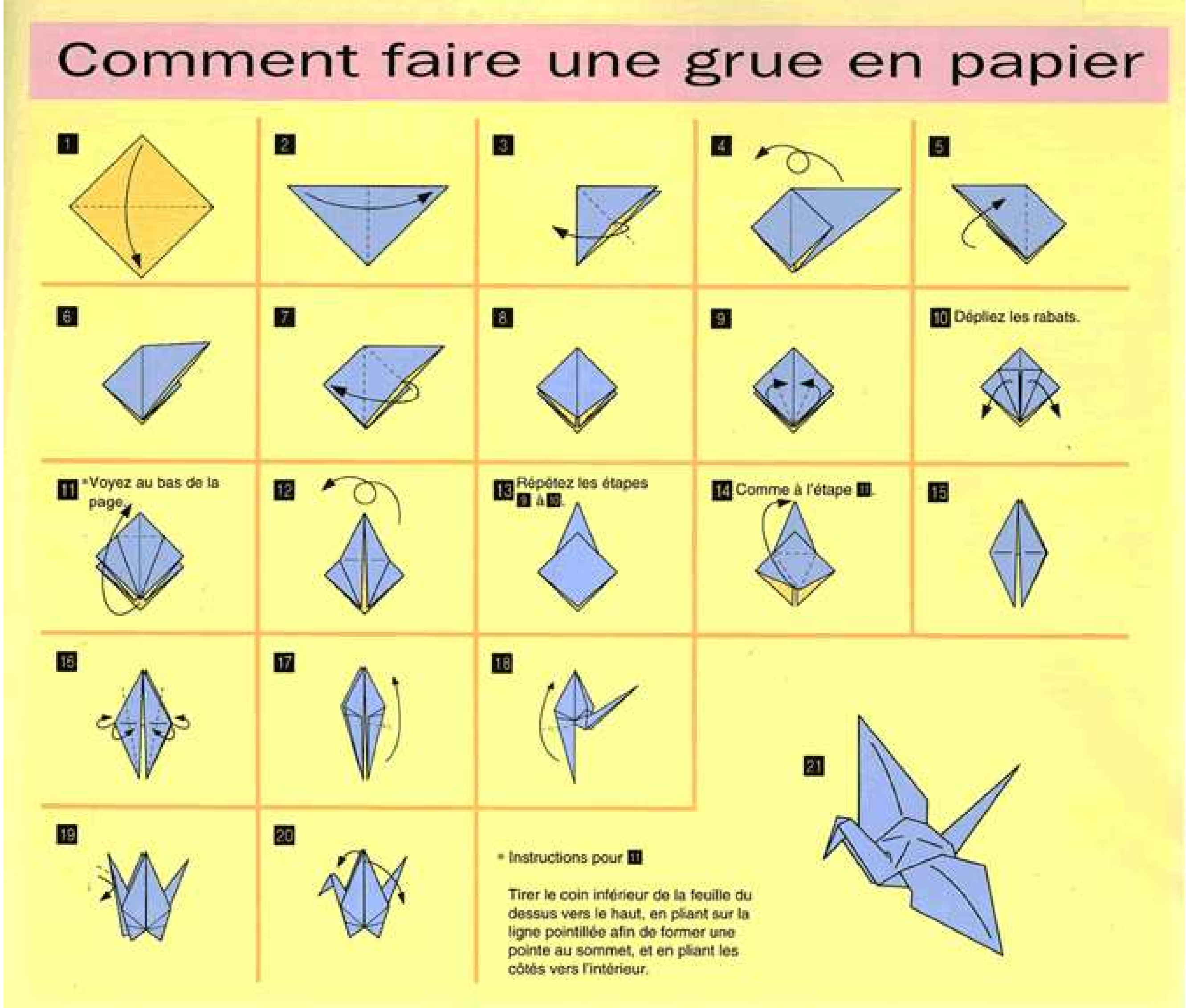
1 maggi cube

2 tsps ginger,garlic and chilli paste

Method

1. Slice onions and keep it aside.
2. Heat oil in a teflon vessel. Put cloves, cinnamon and bay leaves and stir for 30 secs.
3. Put the onions and fry till brown. Add the maggi cube and fry for some time. Put the washed and drained rice and fry for 5 mins. Add the ginger-garlic-chilli paste and saute for some time.
4. Add the hot water and peas. Cover and cook on low fire till done.
5. Serve hot.

Programs



Comment faire une grue en papier

10 Dépliez les rabats.

11 *Voyez au bas de la page.

13 Répétez les étapes 9 à 10.

14 Comme à l'étape 11.

* Instructions pour 11

Tirer le coin inférieur de la feuille du dessus vers le haut, en pliant sur la ligne pointillée afin de former une pointe au sommet, et en pliant les côtés vers l'intérieur.

Programs

- Programs are simple stepwise instructions to complete a task, to be carried out by a person/machine who understands those instructions.
- In different contexts, programs are same as manual, recipes, installation instructions etc.
- In context of instructions to the computers, programs are called algorithms.

Hello World!

A program that prints a line

```
#include <stdio.h>

main()
{

        printf("Hello, World!\n");

}
```

Executing a Program

```
$ cc helloworld.c
$ a.out
Hello, World!
$
```

Human-readable helloworld.c is translated to a.out which is machine-readable.

Simple Arithmetic

```
/* Arithmetic Expressions */

#include <stdio.h>

main()
{

    printf("5 is an integer\n");
    printf("%d is an integer\n",5);

    printf("Approximate value of pi is 22.0/7.0\n");
    printf("Approximate value of pi is %f\n",22.0/7.0);

    printf("100 F corresponds to %f",5.0*(100.0-32.0)/9.0);
}
```

Sequential Programs

The programs are *executed* statement by statement from the top of the file.

Such programs are called *sequential programs*.

There are other possibilities, for example, *parallel programs*

Variables

Variable is a short-form for *variable memory*. Clearly, each variable memorizes/holds/stores a datum (either a number or a character) that can be changed(varied) in the program.

```
main()
{
     int  no_of_apples;

     no_of_apples    = 10;
     printf("I    have   %d  apples\n",    no_of_apples);

     no_of_apples    = 15;
     printf("Now,    I have   %d  apples\n",    no_of_apples);
}
```

Variables

- Variable Name.
- *Data Type* of a variable. `int,float,char` are basic data types.
- *declaration* of a variable

```
int  no_of_students;
float  density_of_mercury;
char  name;
```

- *Assignment* of values to the variables.

```
no_of_students = 18;
density_of_mercury = 13.6;
name = ’c’;
```

Arithmetic with Variables

```
main()
{
    float   p;
    float   r;
    int     t;
    float   i;

    p = 1000.0;
    r = 10.0;
    t = 3;
    i = p * r * t / 100;

    printf("Principle  = %f Rs\n", p);
    printf("Rate       = %f \% pa \n", r);
    printf("Term       = %d years \n", t);
    printf("Interest   = %f Rs \n", i);
}
```

Result

The output:

```
Principle    = 1000.000000    Rs
Rate         = 10.000000    % pa
Term         = 3  years
Interest     = 300.000000    Rs
```

Quadratic Equations

Consider equation

$$Ax^2 + Bx + c = 0.$$

The solution is given by

$$x = \frac{-B \pm \sqrt{B^2 - 4AC}}{2}.$$

We want to write a program such that if A, B and C are given, we get solutions.

Quadratic Equations

- How many variables are required?

Quadratic Equations

- How many variables are required?

  Four. One each for A, B, C and x.

Quadratic Equations

- How many variables are required?

  Four. One each for A, B, C and x.

- What is the data type of these?

Quadratic Equations

- How many variables are required?

  Four. One each for A, B, C and x.

- What is the data type of these?

  Let us assume that all variables are real.

Quadratic Equations

```
#include   <stdio.h>
#include   <math.h>
main()
{
```

Quadratic Equations

```
#include   <stdio.h>
#include   <math.h>
main()
{
     float  A,  B,  C,  x;
```

Quadratic Equations

```
#include    <stdio.h>
#include    <math.h>
main()
{
     float   A,  B,  C,  x;

     A = 1.0;
     B = -4.0;
     C = 3.0;
```

Quadratic Equations

```
#include   <stdio.h>
#include   <math.h>
main()
{
    float  A, B, C, x;

    A = 1.0;
    B = -4.0;
    C = 3.0;

    x = (-B + sqrt(B*B - 4*A*C))/2.0;
    printf("First solution = %f\n", x);
```

Quadratic Equations

```
#include <stdio.h>
#include <math.h>
main()
{
    float A, B, C, x;

    A = 1.0;
    B = -4.0;
    C = 3.0;

    x = (-B + sqrt(B*B - 4*A*C))/2.0;
    printf("First solution = %f\n", x);
    x = (-B - sqrt(B*B - 4*A*C))/2.0;
    printf("Second solution = %f\n", x);
}
```

Quadratic Equations

```
#include   <stdio.h>
#include   <math.h>
main()
{
     float  A,  B,  C,  x;

     scanf("%f",    &A);
     scanf("%f",    &B);
     scanf("%f",    &C);

     x = (-B  + sqrt(B*B   - 4*A*C))/2.0;
     printf("First    solution   = %f\n",   x);
     x = (-B  - sqrt(B*B   - 4*A*C))/2.0;
     printf("Second    solution   = %f\n",   x);
}
```

Quadratic Equations

For any $A, B, C \in \Re$, the solution to

$$Ax^2 + Bx + c = 0$$

is given by (Let $\Delta^2 = B^2 - 4AC$

$$x = \begin{cases} (-B \pm \Delta)/2 & \text{if} \quad \Delta^2 >= 0 \\ (-B \pm i\sqrt{-\Delta^2})/2 & \text{if} \quad \Delta^2 < 0 \end{cases}$$

Quadratic Equations

For any $A, B, C \in \Re$, the solutions to

$$Ax^2 + Bx + c = 0$$

are given by (Let $\Delta^2 = B^2 - 4AC$

$$x = \begin{cases} (-B \pm \Delta)/2 & \text{if} \quad \Delta^2 \geq 0 \\ (-B \pm i\sqrt{-\Delta^2})/2 & \text{if} \quad \Delta^2 < 0 \end{cases}$$

If we want to write this in plain *English*
if $\Delta^2 \geq 0$, then the solutions are $(-B \pm \Delta)/2$

else the solutions are $(-B \pm i\sqrt{-\Delta^2})/2$.

Quadratic Equations

```
main()
{
    float  A,  B,  C,  x,  delta2;

    /*  scanf  etc....   */

    delta2   = B*B  -  4*A*C;

    if  (  delta2   >=  0  )
    {
        x  =  (-B  +  sqrt(delta2))/2.0;
        printf("First     solution    =  %f\n",   x);
        x  =  (-B  -  sqrt(delta2))/2.0;
        printf("Second     solution    =  %f\n",   x);
    }
    else
    {
        delta2   =  sqrt(-delta2)/2;
        x  =  -B/2;
        printf("First     complex   solution    =  (%f,%f)\n",     x,  delta2);
        printf("Second     complex   solution    =  (%f,%f)\n",     x,  -delta2);
    }
}
```

if Statement

The syntax of the **if** statement is

```
if ( condition )
    statement1 ;
else
    statement2 ;
```

- *condition* is some arithmetic or logical condition (true or false)
- *statement1* is called **if-part**
- *statement2* is called **else-part**
- if there are more statements in if-part, they should be grouped inside { and }.
- else-part is optional.

Quadratic Equations

```
main()
{
    float   A,  B,  C,  x,  delta2;                                              |    A    B    C     delta2       x
                                                                                 |
    /*  scanf   etc....   */                                                     |
                                                                                 |    1   -4    3       ?           ?
    delta2   = B*B  - 4*A*C;

    if  (  delta2   >=  0  )
    {
        x  =  (-B  +  sqrt(delta2))/2.0;
        printf("First     solution    =  %f\n",   x);
        x  =  (-B  -  sqrt(delta2))/2.0;
        printf("Second     solution    =  %f\n",   x);
    }
    else
    {
        delta2   =  sqrt(-delta2)/2;
        x  =  -B/2;
        printf("First     complex    solution    =  (%f,%f)\n",     x,  delta2);
        printf("Second     complex    solution    =  (%f,%f)\n",     x,  -delta2);
    }

}
```

Quadratic Equations

```
main()
{
    float   A,  B,  C,  x,  delta2;                           |   A   B   C    delta2     x
                                                              |
    /*  scanf   etc....   */                                  |
                                                              |
    delta2   = B*B  - 4*A*C;                                  |   1  -4    3     4        ?
                                                              |
    if  ( delta2   >= 0 )
    {
        x  = (-B  + sqrt(delta2))/2.0;
        printf("First    solution   = %f\n",   x);
        x  = (-B  - sqrt(delta2))/2.0;
        printf("Second    solution   = %f\n",   x);
    }
    else
    {
        delta2   = sqrt(-delta2)/2;
        x  = -B/2;
        printf("First    complex   solution    = (%f,%f)\n",    x,  delta2);
        printf("Second    complex   solution    = (%f,%f)\n",    x,  -delta2);
    }

}
```

Quadratic Equations

```
main()
{
    float   A,  B,  C,  x,  delta2;                           |   A   B   C    delta2     x
                                                              |
    /*  scanf   etc....   */                                  |
                                                              |
    delta2   = B*B  - 4*A*C;                                  |
                                                              |
    if  (  delta2   >=  0  )                                  |   1  -4     3     4        ?
    {                                                         |   4 >= 0  is true
        x = (-B  + sqrt(delta2))/2.0;
        printf("First    solution   = %f\n",   x);
        x = (-B  - sqrt(delta2))/2.0;
        printf("Second    solution   = %f\n",   x);
    }
    else
    {
        delta2   = sqrt(-delta2)/2;
        x = -B/2;
        printf("First    complex   solution   = (%f,%f)\n",    x,  delta2);
        printf("Second    complex   solution   = (%f,%f)\n",    x,  -delta2);
    }

}
```

Quadratic Equations

```
main()
{
    float   A,  B,  C,  x,  delta2;                           |   A   B   C   delta2      x
                                                              |
    /*  scanf   etc....   */                                  |
                                                              |
    delta2   = B*B  -  4*A*C;                                 |
                                                              |
    if  ( delta2   >=  0 )                                    |
    {                                                         |
        x  =  (-B  +  sqrt(delta2))/2.0;                      |   1  -4    3     4         3
        printf("First     solution    = %f\n",   x);
        x  =  (-B  -  sqrt(delta2))/2.0;
        printf("Second     solution    = %f\n",   x);
    }
    else
    {
        delta2   = sqrt(-delta2)/2;
        x  =  -B/2;
        printf("First     complex    solution    =  (%f,%f)\n",    x,  delta2);
        printf("Second     complex    solution    =  (%f,%f)\n",    x,  -delta2);
    }

}
```

Quadratic Equations

```
main()
{
    float   A,  B,  C,  x,  delta2;                           |   A   B   C    delta2     x
                                                              |
    /*  scanf   etc....   */                                  |
                                                              |
    delta2   = B*B  - 4*A*C;                                  |
                                                              |
    if  ( delta2   >= 0 )                                     |
    {                                                         |
        x = (-B  + sqrt(delta2))/2.0;                         |
        printf("First    solution   = %f\n",   x);            |   1  -4    3     4        3
        x = (-B  - sqrt(delta2))/2.0;
        printf("Second    solution   = %f\n",   x);
    }
    else
    {
        delta2   = sqrt(-delta2)/2;
        x = -B/2;
        printf("First    complex   solution   = (%f,%f)\n",    x,  delta2);
        printf("Second    complex   solution   = (%f,%f)\n",    x,  -delta2);
    }

}
```

Quadratic Equations

```
main()
{
    float   A,  B,  C,  x,  delta2;                            |   A    B    C     delta2       x
                                                               |
    /*  scanf   etc....   */                                   |
                                                               |
    delta2   = B*B  - 4*A*C;                                   |
                                                               |
    if  (  delta2   >=  0  )                                   |
    {                                                          |
        x  =  (-B  +  sqrt(delta2))/2.0;                       |
        printf("First    solution   = %f\n",   x);             |
        x  =  (-B  -  sqrt(delta2))/2.0;                       |   1   -4     3      4          1
        printf("Second    solution   = %f\n",   x);
    }
    else
    {
        delta2   = sqrt(-delta2)/2;
        x  =  -B/2;
        printf("First    complex   solution   =  (%f,%f)\n",    x,  delta2);
        printf("Second    complex   solution   =  (%f,%f)\n",    x,  -delta2);
    }

}
```

Quadratic Equations

```
main()
{
    float  A,  B,  C,  x,  delta2;                            |   A   B   C    delta2      x
                                                              |
    /*  scanf  etc....  */                                    |
                                                              |
    delta2  = B*B - 4*A*C;                                    |
                                                              |
    if ( delta2  >= 0 )                                       |
    {                                                         |
        x = (-B + sqrt(delta2))/2.0;                          |
        printf("First   solution  = %f\n",  x);               |
        x = (-B - sqrt(delta2))/2.0;                          |
        printf("Second   solution  = %f\n",  x);              |   1  -4    3     4         1
    }
    else
    {
        delta2  = sqrt(-delta2)/2;
        x = -B/2;
        printf("First   complex  solution  = (%f,%f)\n",   x,  delta2);
        printf("Second   complex  solution  = (%f,%f)\n",   x,  -delta2);
    }
     /* Execution  jumps  here  */
}
```

Quadratic Equations

```
main()
{
    float  A, B, C, x, delta2;

    /* scanf  etc....  */

    delta2  = B*B - 4*A*C;

    if ( delta2  >= 0 )
    {
        x = (-B + sqrt(delta2))/2.0;
        printf("First    solution   = %f\n",   x);
        x = (-B - sqrt(delta2))/2.0;
        printf("Second    solution   = %f\n",   x);
    }
    else
    {
        delta2  = sqrt(-delta2)/2;
        x = -B/2;
        printf("First    complex   solution   = (%f,%f)\n",    x, delta2);
        printf("Second    complex   solution   = (%f,%f)\n",    x, -delta2);
    }

}
```

A	B	C	delta2	x
1	-4	5	?	?

Quadratic Equations

```
main()
{
    float  A,  B,  C,  x,  delta2;                            |   A   B   C    delta2     x
                                                              |
    /*  scanf  etc....   */                                   |
                                                              |
    delta2  = B*B  - 4*A*C;                                   |   1  -4     5     -4         ?
                                                              |
    if ( delta2  >= 0 )
    {
        x = (-B  + sqrt(delta2))/2.0;
        printf("First   solution  = %f\n",  x);
        x = (-B  - sqrt(delta2))/2.0;
        printf("Second   solution  = %f\n",  x);
    }
    else
    {
        delta2  = sqrt(-delta2)/2;
        x = -B/2;
        printf("First   complex  solution  = (%f,%f)\n",   x,  delta2);
        printf("Second   complex  solution  = (%f,%f)\n",   x,  -delta2);
    }

}
```

Quadratic Equations

```
main()
{
    float  A,  B,  C,  x,  delta2;                                  |   A   B   C    delta2      x
                                                                    |
    /*  scanf  etc....   */                                         |
                                                                    |
    delta2  = B*B  - 4*A*C;                                         |
                                                                    |
    if  ( delta2  >=  0 )                                           |   1  -4    5      -4         ?
    {                                                               |   -4  >=  0   is  false
        x  =  (-B  +  sqrt(delta2))/2.0;
        printf("First    solution   = %f\n",   x);
        x  =  (-B  -  sqrt(delta2))/2.0;
        printf("Second    solution   = %f\n",   x);
    }
    else
    {
        delta2   =  sqrt(-delta2)/2;
        x  =  -B/2;
        printf("First    complex   solution   =  (%f,%f)\n",    x,  delta2);
        printf("Second    complex   solution   =  (%f,%f)\n",    x,  -delta2);
    }

}
```

Quadratic Equations

```
main()
{
    float   A,  B,  C,  x,  delta2;                           |   A   B   C    delta2     x
                                                              |
    /*  scanf   etc....   */                                  |
                                                              |
    delta2   = B*B  - 4*A*C;                                  |
                                                              |
    if  (  delta2   >=  0  )                                  |
    {                                                         |
        x  =  (-B  +  sqrt(delta2))/2.0;
        printf("First     solution    = %f\n",   x);
        x  =  (-B  -  sqrt(delta2))/2.0;
        printf("Second     solution    = %f\n",   x);
    }
    else
    {
        delta2   = sqrt(-delta2)/2;                           |   1  -4     5     1        ?
        x  =  -B/2;
        printf("First     complex   solution    =  (%f,%f)\n",     x,  delta2);
        printf("Second     complex   solution    =  (%f,%f)\n",     x,  -delta2);
    }

}
```

Quadratic Equations

```
main()
{
    float  A,  B,  C,  x,  delta2;                            |   A   B   C    delta2     x
                                                              |
    /* scanf  etc....   */                                    |
                                                              |
    delta2  = B*B  - 4*A*C;                                   |
                                                              |
    if ( delta2  >= 0 )                                       |
    {                                                         |
        x = (-B  + sqrt(delta2))/2.0;
        printf("First    solution   = %f\n",   x);
        x = (-B  - sqrt(delta2))/2.0;
        printf("Second    solution   = %f\n",   x);
    }
    else
    {
        delta2  = sqrt(-delta2)/2;                            |
        x = -B/2;                                             |   1  -4    5     1        2
        printf("First    complex   solution   = (%f,%f)\n",    x,  delta2);
        printf("Second    complex   solution   = (%f,%f)\n",    x,  -delta2);
    }

}
```

Quadratic Equations

```
main()
{
    float  A,  B,  C,  x,  delta2;                                          |   A    B    C     delta2       x

    /*  scanf  etc....  */

    delta2  = B*B  - 4*A*C;

    if  ( delta2  >=  0 )
    {
        x  =  (-B  +  sqrt(delta2))/2.0;
        printf("First   solution  = %f\n",  x);
        x  =  (-B  -  sqrt(delta2))/2.0;
        printf("Second   solution  = %f\n",  x);
    }
    else
    {
        delta2  = sqrt(-delta2);
        x  =  -B/2;
        printf("First   complex  solution  =  (%f,%f)\n",   x,  delta2);
        printf("Second   complex  solution  =  (%f,%f)\n",   x,  -delta2);
    }
     /*  Execution   jumps  here  */
}
```

Conditions

Condition used in **if** is usually a comparison of two numbers or characters using relational operators such as

`<, <=, >, >=, ==, !=`

Here are some examples:

- `no_of_apples > 100`
- `density_of_mercury < 10.0`
- `1 > 0`
- `no_of_boys + no_of_girls < no_of_benches`
- `x == 0.0`
- `rate != 8`

Grading Program

Here is another example: A student in "C Programming"takes 100 marks examination and gets m marks.

And he will get a letter grade according to the following rule:

If $m \geq 80$, then the grade is ’A’

If $m < 80$ AND $m \geq 60$, then the grade is ’B’

If $m < 60$ AND $m \geq 40$, then the grade is ’C’

If $m < 40$, then the grade is ’F’

We want a program, which takes m as input and prints the grade.

Grading Program

```
main()
{
    int  marks;
    char  grade;

    scanf("%d",    &marks);

    if  (  marks   >=  80  )  grade   =  'A';
    if  (  (marks   <  80  )  &&  (marks   >=  60  )  grade   =  'B';
    if  (  (marks   <  60  )  &&  (marks   >=  40  )  grade   =  'C';
    if  (  marks   <  40  )  grade   =  'F';

    printf("Grade    =  %c\n",   grade);
}
```

Grading Program

```
main()
{
    int  marks;
    char  grade;

    scanf("%d",   &marks);

    if ( marks  >= 80 ) grade  = ’A’;
    else
    {
       if ( marks  >= 60 ) grade  = ’B’;
       else
       {
           if ( marks  >= 40 ) grade  = ’C’;
           else  grade  = ’F’;
       }
    }

    printf("Grade   = %c\n",  grade);
}
```

E

xercise: Given three integers a, b, and c, write a program to print these in ascending order.

For Example, if $a = 5$, $b = 1$ and $c = 3$, output should be 1, 3, 5.