CSE 113 Structured Programming Language

Input/Output functions & Control Statements

Tasnim Zahan
Assistant Professor
Dept. of CSE
NEUB

Operator Precedence

Table 3.1 Operator Precedence Groups

Operator category	*Operators*	*Associativity*
unary operators	`- ++ -- ! sizeof (type)`	R → L
arithmetic multiply, divide and remainder	`* / %`	L → R
arithmetic add and subtract	`+ -`	L → R
relational operators	`< <= > >=`	L → R
equality operators	`== !=`	L → R
logical *and*	`&&`	L → R
logical *or*	`\\`	L → R
conditional operator	`? :`	R → L
assignment operators	`= += -= *= /= %=`	R → L

A complete listing of all C operators, which is more extensive than that given in Table 3.1, is shown in Appendix C.

Exercises

Consider a=15, b = 5, f = 1.5, ch = ‘B’. Find out the value of the following expressions:

1. a -= (b%3) // a = a - (b%3)
2. ((f!=b) && (ch<=‘B’)) || ((a*2)>=26)
3. f *= b // f = f * b
4. ch != ‘d’

Library Functions

Table 3.2 Some Commonly Used Library Functions

Function	*Type*	*Purpose*
`abs(i)`	int	Return the absolute value of `i`.
`ceil(d)`	double	Round up to the next integer value (the smallest integer that is greater than or equal to `d`).
`cos(d)`	double	Return the cosine of `d`.
`cosh(d)`	double	Return the hyperbolic cosine of `d`.
`exp(d)`	double	Raise *e* to the power `d` (*e* = 2.7182818... is the base of the natural (Naperian) system of logarithms).
`fabs(d)`	double	Return the absolute value of `d`.
`floor(d)`	double	Round down to the next integer value (the largest integer that does not exceed `d`).
`fmod(d1,d2)`	double	Return the remainder (i.e., the noninteger part of the quotient) of `d1/d2`, with same sign as `d1`.
`getchar( )`	int	Enter a character from the standard input device.
`log(d)`	double	Return the natural logarithm of `d`.
`pow(d1, d2)`	double	Return `d1` raised to the `d2` power.
`printf(...)`	int	Send data items to the standard output device (arguments are complicated —see Chap. 4).
`putchar(c)`	int	Send a character to the standard output device.
`rand( )`	int	Return a random positive integer.
`sin(d)`	double	Return the sine of `d`.
`sqrt(d)`	double	Return the square root of `d`.
`srand(u)`	void	Initialize the random number generator.
`scanf(...)`	int	Enter data items from the standard input device (arguments are complicated —see Chap. 4).
`tan(d)`	double	Return the tangent of `d`.
`toascii(c)`	int	Convert value of argument to ASCII.
`tolower(c)`	int	Convert letter to lowercase.
`toupper(c)`	int	Convert letter to uppercase.

the data type of the quantity that is returned by the function.

Input & output functions

Input functions	Output functions
getchar();	putchar();
gets();	puts();
scanf();	printf();

Input & output functions

- getchar(): reads only a single character (input)
- putchar(): writes only a single character (output)

```
#include <stdio.h>
int main(){
   char ch;

   printf("Enter a character: ");
   ch = getchar();

   printf("\nYou entered: ");
   putchar(ch);

   return 0;
}
```

Input & output functions

- gets(): reads a line of text (string)
- puts(): writes a line of text (string)

```
#include <stdio.h>
int main(){
   char str[100];  // string

   printf("Enter a line of text:\n");
   gets(str);

   printf("\nYou entered: ");
   puts( str );

   return 0;
}
```

Input & output functions

- scanf(): reads the input according to the format provided
- printf(): produces the output according to the format provided

```
#include <stdio.h>
int main(){
   int i;
   float x;

   printf("Enter an integer and a float:");
   scanf("%d %f", &i, &x);

   printf( "\nYou entered: %f %d ", x, i);

   return 0;
}
```

The format can be a simple constant string, but we can specify %d, %f, %c, %s etc.., to print or read integer, float, character or string respectively

Input & output functions

- String input, output using scanf() and printf()

```
#include <stdio.h>
int main(){
   char word[30], line[100];

   printf("Enter a word: ");
   scanf("%s", word);

   printf("Enter a line of text: ");
   scanf("%[^\n]", line);

   printf("you entered:\n%s\n%s", word, line);
   return 0;
}
```

The format can be a simple constant string, but we can specify %d, %f, %c, %s etc.., to print or read integer, float, character or string respectively

Assignment

1. Write a C program to prompt the user to input 3 integer values and print these values in forward and reversed order.
2. Write a program to check odd or even number using modulus operator.
3. Print the value of y for given x=3 & z=6

```
a) y = x++ + ++x; //x = x + 1
b) y = ++x + ++z;
c) y = x>z? x:z;
```

Condition Logic: if

```
if (test expression)
{
        /* statements to be
        executed if the test
        expression is true*/
}
```

Expression is true.

```
int test = 5;

if (test < 10)
{
  // codes
}

// codes after if
```

Expression is false.

```
int test = 5;

if (test > 10)
{
  // codes
}

// codes after if
```

Condition Logic: if

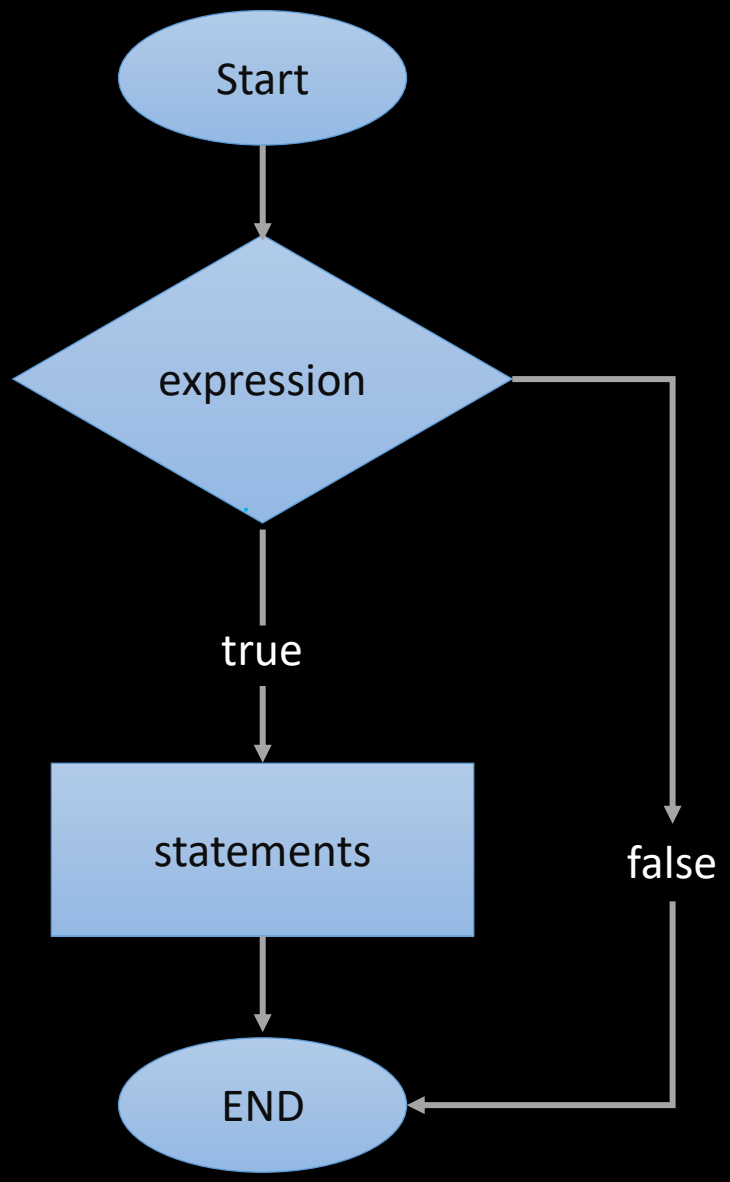
num 45

```
#include <stdio.h>
int main() {
    int num;
    printf("Enter an integer: ");
    scanf("%d", &num);

    // true if number is less than 0
    if (number < 0) {
        printf("You entered %d\n", number);
    }
    printf("The End");
    return 0;
}
```

Condition Logic: if-else

```
if(expression)
{
    statement 1;
    statement 2;
    ...
}
else{
    statement 1;
    statement 2;
    ...
}
```



Condition Logic: if-else

```
if(expression)
{
    statement 1;
    statement 2;
    ...
}
else{
    statement 1;
    statement 2;
    ...
}
```

Expression is true.

```
int test = 5;

if (test < 10)
{
   // body of if
}
else
{
   // body of else
}
```

Expression is false.

```
int test = 5;

if (test > 10)
{
  // body of if
}
else
{
  // body of else
}
```

Even or Odd

```
#include <stdio.h>
int main(){
   int num;
   printf("Enter a number\n");
   scanf("%d", &num);

   if(num%2 == 0){
      printf("Even number\n");
   }
   else{
      printf("Odd number\n");
   }


   return 0;
}
```

Positive or Negative

```
#include <stdio.h>
int main(){
   int num;
   printf("Enter a number\n");
   scanf("%d", &num);

   if(num >= 0){
      printf("Positive number\n");
   }
   else{
      printf("Negative number\n");
   }


   return 0;
}
```

Nested if

```
if(condition){
    statement 1;
    ...
    if(expression){

    }
    else{

    }
}
else{
    statement 1;
    ...
}
```

```
if(condition){
    statement 1;
    ...
}
else if(condition){
    statement 1;
    ...
}
else{
    statement 1;
    ...
}
```

Positive or Negative or Zero

```
#include <stdio.h>
int main(){
   int num;
   printf("Enter a number\n");
   scanf("%d", &num);

   if(num > 0){
      printf("Positive number\n");
   }
   else if (num == 0){
      printf("Zero\n");
   }
   else{
      printf("Negative number\n");
   }


   return 0;
}
```

```c
#include <stdio.h>
int main() {
    int number1, number2;
    printf("Enter two integers: ");
    scanf("%d %d", &number1, &number2);

    if (number1 >= number2) {
      if (number1 == number2) {
        printf("Result: %d = %d",number1,number2);
      }
      else {
        printf("Result: %d > %d", number1, number2);
      }
    }
    else {
        printf("Result: %d < %d",number1, number2);
    }

    return 0;
}
```

Output?

num1 = 5
num2 = 12

```
#include <stdio.h>
int main(){
    int num1, num2;
    printf("Enter two numbers\n");
    scanf("%d%d", &num1, &num2);

    if(num1 > num2){
        printf("First number is bigger\n");
    }
    else if (num1 < num2 ){
        printf("Second number is bigger\n");
    }
    else{
        printf("Two numbers are equal\n");
    }

    return 0;
}
```

Output?

num1 = 12
num2 = 12

```
#include <stdio.h>
int main(){
    int num1, num2;
    printf("Enter two numbers\n");
    scanf("%d%d", &num1, &num2);

    if(num1 >= num2){
        printf("First number is bigger\n");
    }
    if (num1 < num2 ){
        printf("Second number is bigger\n");
    }
    if (num1 == num2){
        printf("Two numbers are equal\n");
    }

    return 0;
}
```

Problem statement

- Write a C program to prompt the user to input 3 integer values and print the largest number.
- Take a character as input and check for ‘r’, ‘g’, ‘b’.
  - If ‘r’ print ‘RED’
  - If ‘g’ print ‘GREEN’
  - if ‘b’ print ‘BLUE’
  - Otherwise print ’Not Matched’