

0.1 ICS 104 - Introduction to Programming in Python and C

0.2 Overview of C - Lab 2

0.3 Online C Compilers

https://www.onlinegdb.com/online_c_compiler (https://www.onlinegdb.com/online_c_compiler)

<https://www.codechef.com/ide> (<https://www.codechef.com/ide>)

1 Lab Learning Outcomes

- To develop code using if statements in C.
- To develop loops in C.

1.0.1 Compound or block statement

A Compound statement consists of one or several individual statements enclosed within a pair of curly braces { }. The individual statements may themselves be simple statements or compound statements. A compound statement is treated by the compiler as if it were a single statement. Unlike simple statements, a compound statement does not end with a semicolon.

Example:

```
{  
  
    printf("Enter the radius [cm]: ");  
  
    scanf("%lf", &radius);  
  
    printf("Circle area = %0.2f\n", PI * radius * radius);  
  
}
```

1.0.2 One-way selection (if-statement)

Used to execute a statement or a compound_statement when a condition is true.

Simple statement	Compound statement
<pre>if (condition) statement1;</pre>	<pre>if(condition) compound_statement1</pre>

Examples:

<pre>printf("Enter a number: "); double num; scanf("%lf", &num); if (num > 0) printf("The square root is %f", sqrt(num));</pre>	<pre>printf("Enter a number: "); double num; scanf("%lf", &num); if (num > 0) { printf("The square root is %f", sqrt(num)); printf("The natural logarithm is %f", log(num)); }</pre>
---	--

1.0.3 Two-way selection (if-else statement)

An if-else statement is used to execute a statement or a compound-statement when a condition is true; and another statement or compound-statement when that condition is false.

Simple statements	Compound statements
<pre>if (condition) statement1; else statement2;</pre>	<pre>if (condition) compound_statement1 else compound_statement2</pre>

Example: Finding max of two numbers

```

if(x >= y)
    max = x;
else
    max = y;

```

Example:

```

char currencyType;
scanf("%c", &currencyType);
if(currencyType == 's' || currencyType == 'S'){
    printf("Enter positive amount ");
    scanf("%lf", &amount);
    riyalBalance = riyalBalance + amount;
}
else
    printf("Wrong currency type");

```

1.0.4 Multi-way selection with an else option (if- else if - else statement)

Used to execute the first statement or the first compound_statement whose corresponding condition is true. The statement in the else part is executed if each condition is false.

Simple statements	Compound statements
<pre> if(condition1) statement1; else if(condition2) statement2; else if(condition3) statement3; . . . else if(conditionM) statementM; else statementN; </pre>	<pre> if(condition1) compound_statement1 else if(condition2) compound_statement2 else if(condition3) compound_statement3 . . . else if(conditionM) compound_statementM else compound_StatementN </pre>
<p>Note: There may be one or more <i>else if</i> branches</p>	

```
In [ ]: 1 Example: Classify a numeric grade:
2
3 int validGrade = 1; // initialize validGrade to true
4 double grade;
5 char letterGrade;
6
7 printf("Enter grade");
8 scanf("%lf", &grade);
9
10 if(grade < 0.0 || grade > 100.0)
11     validGrade = 0; // set validGrade to false
12 else if(grade >= 85.0)
13     letterGrade = 'A';
14 else if(grade >= 75.0)
15     letterGrade = 'B';
16 else if(grade >= 65.0)
17     letterGrade = 'C';
18 else if(grade >= 45.0)
19     letterGrade = 'D';
20 else
21     letterGrade = 'F';
22
23 if(validGrade)
24     printf("The letter grade for %0.1f is %c", grade, letterGrade);
25 else
26     printf("Error: Invalid grade");
27
```

1.0.5 Multi-way selection without an else option (if - else if - else if statement)

Used to execute the first statement or compound_statement whose corresponding condition is true. No if-branch is executed if each condition is false.

Single statements	Compound statements
<pre> if(condition1) statement1; else if(condition2) statement2; else if(condition3) statement3; . . . else if(conditionM) statementM; else if(conditionN) statementN; </pre>	<pre> if(condition1) compound_statement1 else if(condition2) compound_statement2 else if(condition3) compound_statement3 . . . else if(conditionM) compound_statementM else if(conditionN) compound_StatementN </pre>
<p>Note: There may be one or more <i>else if</i> branches</p>	

In []:

```

1 Example:
2
3 if(grade >= 0 && grade < 45)
4     printf("Fail");
5 else if(grade >= 45 && grade <= 100)
6     printf("Pass");
7

```

1.0.6 Nested if statements

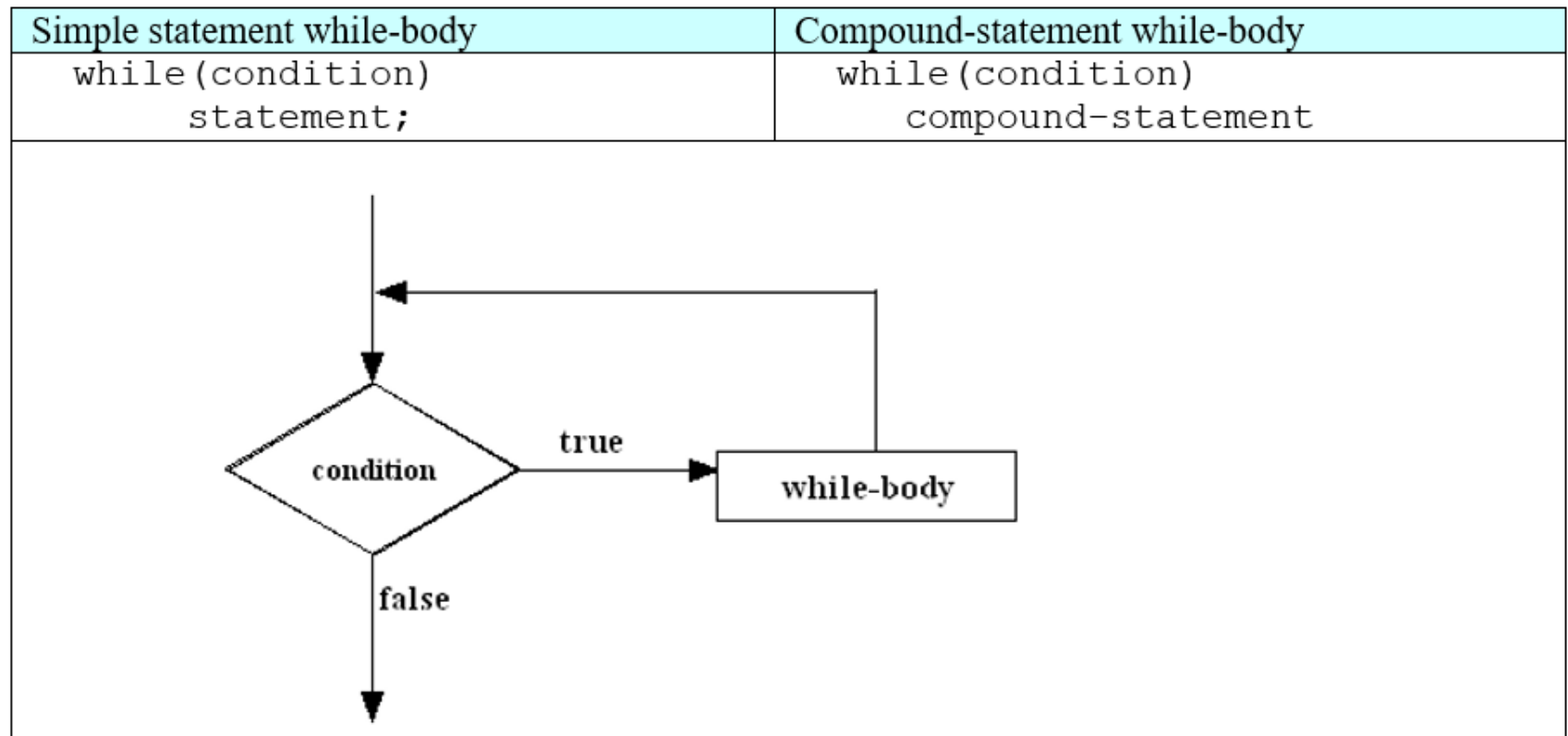
The compound statement in an if-branch or an else-branch of an if-statement may contain one or more of any type of if-statement discussed in the previously.

In []:

```
1 Example:
2     if(grade < 0.0 || grade > 100.0)
3         printf("Error: Invalid grade");
4     else{
5         if(grade >= 85.0)
6             letterGrade = 'A';
7         else if(grade >= 75.0)
8             letterGrade = 'B';
9         else if(grade >= 65.0)
10            letterGrade = 'C';
11        else if(grade >= 45.0)
12            letterGrade = 'D';
13        else
14            letterGrade = 'F';
15
16        printf("The letter grade for %0.1f is %c", grade, letterGrade);
17    }
18
```

1.0.7 while-statement

The while loop has the format shown below. If the loop body has more than one statement, then they must be put between curly brackets {}.



Examples:

while-loop	output
<pre> int n = 1; // initialization while(n <= 10){ // test printf("%d ", n); n = n+2; // update } </pre>	1 3 5 7 9
<pre> int k = 12; // initialization while(k > 6) { // test printf("%d ", k); k=k-1; // update } </pre>	12 11 10 9 8 7

```
In [ ]: 1 Example: write a program that reads n floating point numbers and then it finds their sum, product, and average.
2
3 #include <stdio.h>
4 int main(void){
5     int n ,i;
6     double value, sum,product,average;
7     printf("Enter number of values to process: ");
8     scanf("%d", &n); // assume n > 0
9     sum = 0;
10    product = 1;
11    i = 1; // initialization
12    while(i <= n){ // test
13        printf("Enter value %d: ",i);
14        scanf("%lf",&value);
15        sum = sum + value;
16        product = product * value;
17        i++; // update
18    }
19
20    average = sum / n;
21    printf("Sum = %.2f\n",sum);
22    printf("Product= %.2f\n",product);
23    printf("Average= %.2f\n",average);
24    return 0;
25 }
```

1.0.8 Sentinel controlled loops

In a program, a sentinel is a value that marks the end of a series of data values; but is not a data value itself.

Sentinels may be used to control conditional loops:

```
In [ ]: 1 Example: Write a C program fragment that prompts for and reads student grades in a quiz. It then calculates
        2 Use a negative value or a value > 100 as the sentinel.
        3
        4 int count = 0;
        5 double grade, sumOfGrades = 0.0;
        6 printf("Enter grade#%d (-ve value or value > 100 to terminate)\n", count+1);
        7 scanf("%lf", &grade);
        8 while(grade >= 0 && grade <= 100){
        9     count++;
        10    sumOfGrades += grade;
        11    printf("Enter grade#%d (-ve value or value > 100 to terminate)\n", count+1);
        12    scanf("%lf", &grade);
        13 }
        14 if(count == 0)
        15     printf("Error: No valid grade entered\n");
        16 else
        17     printf("Average = %.2f\n", sumOfGrades / count);
        18
```

1.0.9 do-while statement

A do-while statement is used to execute a statement or a compound-statement one or more times as long as the do-while condition is true:

```
do
    do-while body
while(condition);
```

The do-while body can be a simple statement in which case it must be terminated by a semicolon or it may be a compound-statement in which case it MUST NOT be terminated by a semicolon:

Simple statement do-while body	Compound-statement do-while body
<pre>do statement; while (condition);</pre>	<pre>do compound_statement while (condition);</pre>
<pre>graph TD; Entry(()) --> Body[do-while body]; Body --> Condition{condition}; Condition -- true --> Entry; Condition -- false --> Exit(());</pre>	

Note: Because the condition of a do-while loop is at the end; a do-while loop executes one or more times.

Examples:

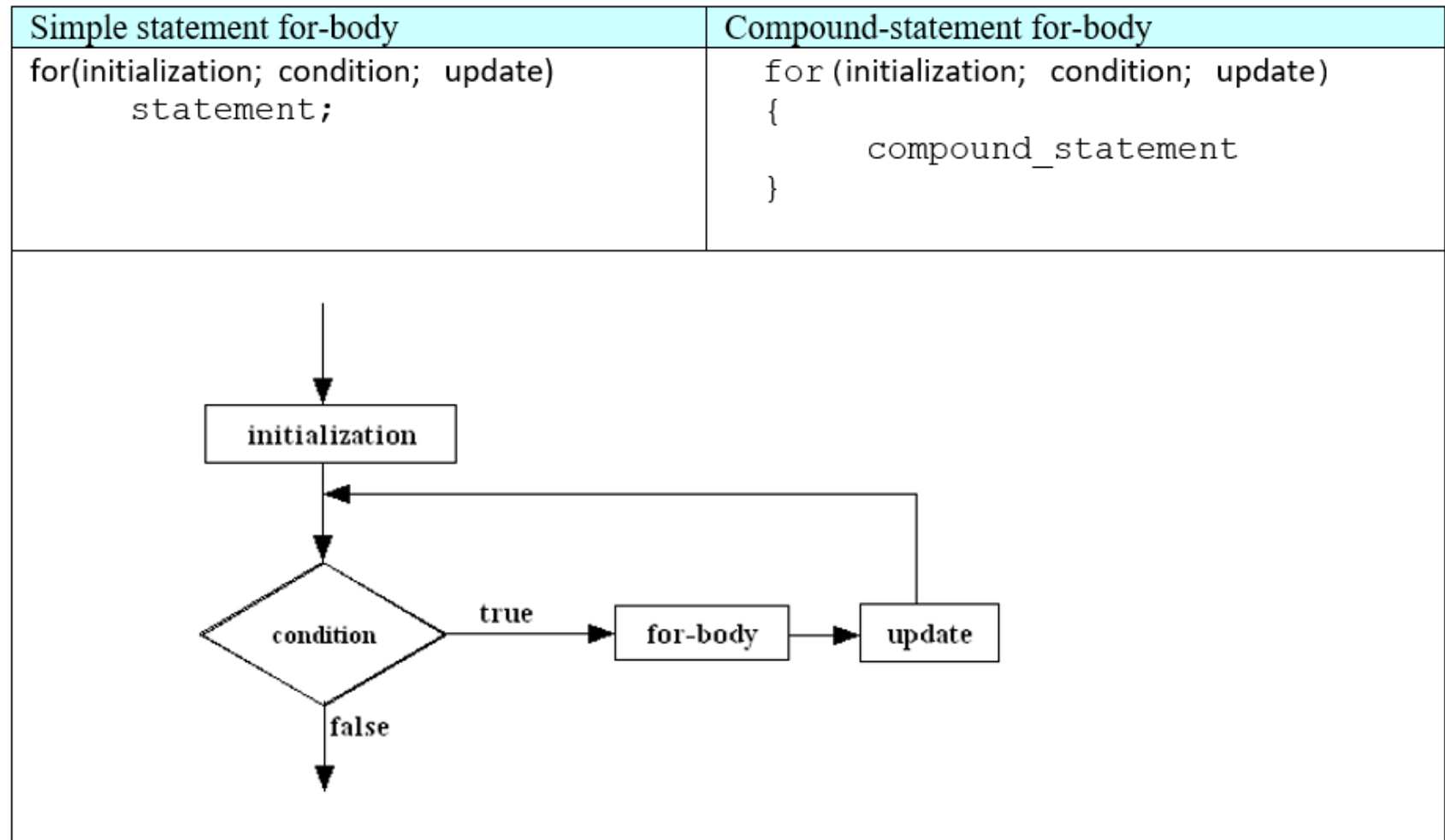
do-while loop	output
<pre>int n = 1; do{ printf("%d ", n); n += 2; } while(n <= 10);</pre>	1 3 5 7 9
<pre>int x = 25; do printf("%d ", x -= 5); while(x > 0);</pre>	20 15 10 5 0

```
In [ ]: 1 Example: A do-while loop can be used to validate input
2
3     int n;
4     do{
5         printf("Enter an integer number in the [10,100] interval: \n");
6         scanf("%d", &n);
7         if(n < 10 || n > 100)
8             printf("Sorry wrong input, try again\n");
9     }while (n < 10 || n > 100);
10    printf("Now your input is correct\n");
```

1.0.10 for-statement

In a for-loop the 3 steps of initialization, test (condition) and update are grouped in one place as shown below. Similar to while loop, if the for loop body has more than one statement, then they must be put between curly brackets { }.

```
for(initialization; condition; update)
    for-loop-body
```



Examples:

for-loop	output
<pre>int k; for(k = 7; k <= 12; k++) printf("%d ", k);</pre>	7 8 9 10 11 12
<pre>int x ; for(x = 8; x > 2; x=x-1) if(x % 2 == 0) printf("%d is even\n", x); else printf("%d is odd\n", x);</pre>	8 is even 7 is odd 6 is even 5 is odd 4 is even 3 is odd

1.0.11 Nested loops

A loop statement may contain in its body one or more loop statements. There are two types of nested loops: independent and dependent. A nested loop is independent if its number of repetitions does not depend on an outer loop. A nested loop is dependent if its number of repetitions depends on an outer loop.

Example:

Nested loops	output
<pre>// Example of independent nested //loop int m, n; for(m = 5; m >= 1; m--){ printf("m is now %d\n", m); for(n = 1; n <= 4; n++) printf("n = %d ", n); printf("\n"); }</pre>	<pre>m is now 5 n = 1 n = 2 n = 3 n = 4 m is now 4 n = 1 n = 2 n = 3 n = 4 m is now 3 n = 1 n = 2 n = 3 n = 4 m is now 2 n = 1 n = 2 n = 3 n = 4 m is now 1 n = 1 n = 2 n = 3 n = 4</pre>
<pre>//Example of dependent nested loop int k,m; for(k=1;k<=9;k++){ for(m = 1; m <= k; m++){ printf("%d",k); } printf("\n"); }</pre>	<pre>1 22 333 4444 55555 666666 7777777 88888888 999999999</pre>

```
In [ ]: 1 Example: Write a C program that prompts for and reads three quiz grades for each student in a class of four
2 The program then computes and displays the average for each student. Your program must be easily modifiable
3 any number of students and quizzes.
4
5 #include <stdio.h>
6 #include <stdlib.h>
7 #define NUMSTUDENTS 4
8 #define NUMQUIZES 3
9 int main(void){
10     double grade, studentTotal, studentAverage;
11     int m, n;
12     for(m = 1; m <= NUMSTUDENTS; m++){
13         studentTotal = 0.0;
14         for(n = 1; n <= NUMQUIZES; n++){
15             printf("Enter QuizGrade%d for student%d\n", n, m);
16             scanf("%lf", &grade);
17             studentTotal += grade;
18         }
19         studentAverage = studentTotal / NUMQUIZES;
20         printf("The average for student%d is %.2f\n", m, studentAverage);
21     }
22
23     return 0;
24 }
```

2 Exercises

2.1 Exercise # 1:

Write an interactive C program that displays the following menu:

1. Find area of a triangle.
 2. Find volume and surface area of a sphere.
- Please select your choice (1 or 2):

The program then reads the menu choice and behaves as in the following table:

Menu choice	Program behavior
Input other than 1, and 2	The program displays the following error message : Error: Wrong menu choice and then terminates.
1	<p>The program prompts for and reads the lengths of the three sides of a triangle. It then computes and displays the area the triangle.</p> <p>Given a triangle with sides a, b, and c, Heron's formula for the triangle area is:</p> $S = (a + b + c) / 2$ $Area = \sqrt{S(S - a)(S - b)(S - c)}$ <p>Your program must display the error message: Error: Invalid triangle. and terminate, if any of the following conditions is satisfied:</p> $c \geq a + b$ $a \geq b + c$ $b \geq a + c$
2	<p>The program prompts for and reads the radius r of a sphere. It then computes and displays the volume and the surface area of the sphere. [$Volume = \frac{4}{3}\pi r^3$, $Surface Area = 4\pi r^2$]</p> <p>$\pi = 3.14159$</p> <p>Your program must display the error message: Error: Invalid radius value. and terminate, if $r \leq 0$</p>

Note:

- Assume that the values read by the program for options 1, and 2 are in centimeters. For these options, your program must display appropriate units in the output.
- Solve the problem by using multi-way if. DO NOT USE SINGLE-WAY if.

Sample program runs:

```
1. Find area of a triangle.  
2. Find volume and surface area of a sphere.  
Please select your choice (1 or 2): 6
```

```
Error: Wrong menu choice
```

```
1. Find area of a triangle.  
2. Find volume and surface area of a sphere.  
Please select your choice (1 or 2): 1  
Enter the lengths [cm] of the three sides of a triangle: 1 2 3  
Error: Invalid triangle.
```

```
1. Find area of a triangle.  
2. Find volume and surface area of a sphere.  
Please select your choice (1 or 2): 1  
  
Enter the lengths [cm] of the three sides of a triangle: 3.0 4.0 5.0  
  
Triangle area = 6.00 square cm
```

```
1. Find area of a triangle.  
2. Find volume and surface area of a sphere.  
Please select your choice (1 or 2): 2  
  
Enter the radius [cm] of a sphere: -8.5  
Error: Invalid radius.
```

```
1. Find area of a triangle.  
2. Find volume and surface area of a sphere.  
Please select your choice (1 or 2): 2  
  
Enter the radius [cm] of a sphere: 4.5
```

Sphere volume = 381.70 cubic cm, Sphere surface area = 254.47 square cm

In []:

```
1 // your code
2
3 #include <stdio.h>
4 #include <math.h>
5 #define PI 3.14159
6
7 int main()
8 {
9     int choice;
10    double triangle,side1,side2,side3,area,sphereVoulume,radius,surfaceArea;
11    printf("1.Find area of a triangle.\n");
12    printf("2.Find volume and surdace area of a sphere.\n");
13    printf("Please select Your choice (1 or 2): ");
14    scanf("%d", &choice);
15
16    if(choice == 1) {
17        printf("Enter the lengths [cm] of three sides of a triangle: ");
18        scanf("%lf %lf %lf", &side1,&side2, &side3);
19        if(side3 >= side1 + side2 || side2 >= side1 + side3 || side1 >= side2 + side3) {
20            printf("Error: Invalid triangle.");
21        }
22        else {
23            triangle = (side1 + side2 + side3) / 2;
24            area = sqrt(triangle * (triangle - side1) * (triangle - side2) * (triangle - side3));
25            printf("Triangle area = %.2lf ", area);
26        }
27    }
28
29    else if(choice == 2) {
30        printf("Enter the radius [cm] of a sphere: ");
31        scanf("%lf", &radius);
32        if (radius <= 0){
33            printf("Invalid radius value.");
34        }
35        else {
36            sphereVoulume = ((4 * PI * pow(radius,3))/3);
37            surfaceArea = (4 * PI * pow(radius,2));
38            printf("Sphere volume = %.2lf cubic cm , Sphere surface area = %.2lf square cm", sphereVoulume, surfaceArea);
39        }
40    }
41    else {
42        printf("Wrong Menu Choice.");
```

```

43 }
44
45
46 return 0;
47 }

```

2.2 Exercise # 2:

An n-digit positive integer x greater than 0:

satisfies property A, if:

$$d_1 d_2 d_3 \dots d_n = d_1^n + d_2^n + d_3^n + \dots + d_n^n$$

For example, each of the numbers 4, 153, 371, 9474, 54748, and 548834 satisfies property A:

$$4 = 4^1$$

$$153 = 1^3 + 5^3 + 3^3$$

$$371 = 3^3 + 7^3 + 1^3$$

$$9474 = 9^4 + 4^4 + 7^4 + 4^4$$

$$54748 = 5^5 + 4^5 + 7^5 + 4^5 + 8^5$$

$$548834 = 5^6 + 4^6 + 8^6 + 8^6 + 3^6 + 4^6$$

Write an interactive C program that prompts for and reads a positive integer n in the range 1 to 99999 inclusive. Your program must loop as long as the input n is invalid. If the input is valid, the program determines whether the integer n satisfies property A or not.

Sample program runs:

```
Enter a positive integer > 0: 153  
153 satisfies property A.
```

```
Enter a positive integer > 0: -5  
Error: Invalid input.  
Enter a positive integer > 0: 70000000  
Error: Invalid input.  
Enter a positive integer > 0: 568  
568 does not satisfy property A.
```

```
Enter a positive integer > 0: -75  
Error: Invalid input.  
Enter a positive integer > 0: 54748  
54748 satisfies property A.
```

```
In [ ]: 1 // your code
2
3
4 #include<stdio.h>
5 #include<math.h>
6
7 int main(){
8 int number, digit, count=0, sum=0;
9 do{
10 printf("Enter a poistive integer > 0: ");
11 scanf("%d", &number);
12 if(number >= 1 && number <= 99999){
13 digit = number;
14 while(digit){
15 digit /= 10;
16 count++;
17 }
18 digit = number;
19 while(digit){
20 sum += pow(digit%10, count);
21 digit /= 10;
22 }
23 if(number != sum)
24 printf("%d does not satisfy property A.\n", number);
25 else printf("%d satisfies property A.\n", number);
26 }
27 else printf("Error: Invalid input.\n");
28 }while(!(number >= 1 && number <= 99999));
29 return 0;
30 }
```

2.3 Exercise # 3:

Write a C program that uses appropriate nested loops to generate the following multiplication table:

1 * 1 = 1	1 * 2 = 2	1 * 3 = 3	1 * 4 = 4	1 * 5 = 5
2 * 1 = 2	2 * 2 = 4	2 * 3 = 6	2 * 4 = 8	2 * 5 = 10
3 * 1 = 3	3 * 2 = 6	3 * 3 = 9	3 * 4 = 12	3 * 5 = 15
4 * 1 = 4	4 * 2 = 8	4 * 3 = 12	4 * 4 = 16	4 * 5 = 20
5 * 1 = 5	5 * 2 = 10	5 * 3 = 15	5 * 4 = 20	5 * 5 = 25
6 * 1 = 6	6 * 2 = 12	6 * 3 = 18	6 * 4 = 24	6 * 5 = 30
7 * 1 = 7	7 * 2 = 14	7 * 3 = 21	7 * 4 = 28	7 * 5 = 35

In []:

```

1 // your code
2
3 #include <stdio.h>
4
5 int main()
6 {
7     int i,j;
8     for(i=1; i<=7; i++)
9     {
10        for(j=1; j<=5; j++)
11        {
12            printf("%d * %d = %d\t", i, j, i*j);
13        }
14        printf("\n");
15    }
16    return 0;
17 }

```