To:
Tenouk

C LAB WORKSHEET 8a_1
C & C++ Selection: More on C/C++ if and if-else Part 4

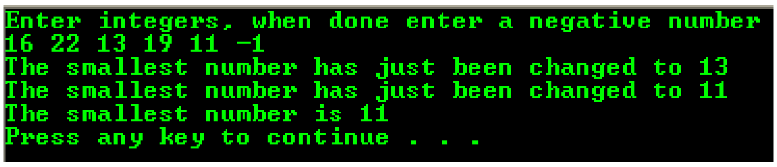
Items in this page:

1. More on if, if-else variations and flowcharts.
2. Activities, questions and answers.
3. Tutorial reference that should be used together with this worksheet are: C & C++ program control 1 and C & C++ program control 2.

14. Enter a sample input: 16, 22, 13, 19, 11, -1 for the following experiment.

```
#include <stdio.h>

int main(void)
{
      int k, smallest;
      printf("Enter integers, when");
      printf(" done enter a ");
      printf("negative number\n");
      scanf_s("%d", &k, 1);
      // assign the first number to smallest variable
      smallest = k;
      // iterate while k >= 0
      for( ; k >= 0; )
      {
            // if the entered number is < smallest
            if(k < smallest)
            {
                  // then assign the number to smallest variable...
                  smallest = k;
                  // do some checking...
                  printf("The smallest number has just been changed to %d\n", smallest);
            }
            // read the next input....repeat
            scanf_s("%d", &k, 1);
      }
      // print the smallest number...
      printf("The smallest number is %d\n", smallest);
      return 0;
}
```

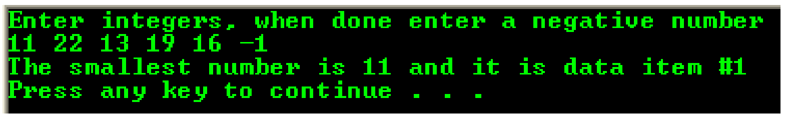
a. Draw the flowchart.
b. When running this program, how many times smallest changed?
c. When running this program with the following data: 11, 22, 13, 19, 16, -1, how many times did smallest change?
d. Are the braces lined up with the if statement necessary to give the same result?


```
Enter integers, when done enter a negative number
16 22 13 19 11 -1
The smallest number has just been changed to 13
The smallest number has just been changed to 11
The smallest number is 11
Press any key to continue . . .
```

b. 3 times.
c. 0 times, because the smallest number is the first entered data.
d. Yes.

```
#include <stdio.h>

int main(void)
{
    int k, i, item_num = 1, smallest;
    printf("Enter integers, when");
    printf(" done enter a ");
    printf("negative number\n");
    scanf_s("%d", &k, 1);
    // assign the first number to smallest variable
    smallest = k;
    // iterate while k >= 0
    for(i=1; k >= 0; i++)
    {
        // if the entered number is < smallest
        if(k < smallest)
        {
            // then assign the number to smallest variable...
            smallest = k;
            item_num = i;
            // do some checking...
            printf("The smallest number has just been changed to %d\n", smallest);
         }
         // read the next input....repeat
         scanf_s("%d", &k, 1);
    }
    // print the smallest number...
    printf("The smallest number is %d and it is data item #%d\n", smallest, item_num);
    return 0;
}
```

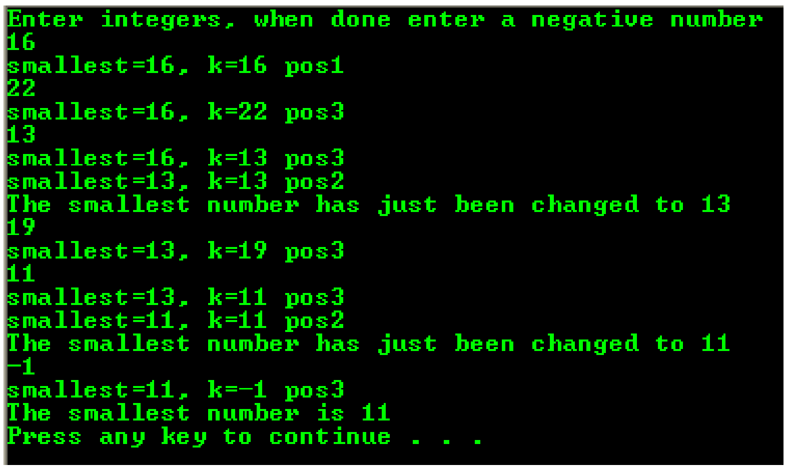
e. Now try to alter the program so that it also prints the data item that was entered. For example, with the following data: 11, 22, 13, 19, 16, -1, this should be printed: “The smallest is 11 and it was data item number 1.” If you can’t do it, don’t be concerned. The solution is in the next experiment.



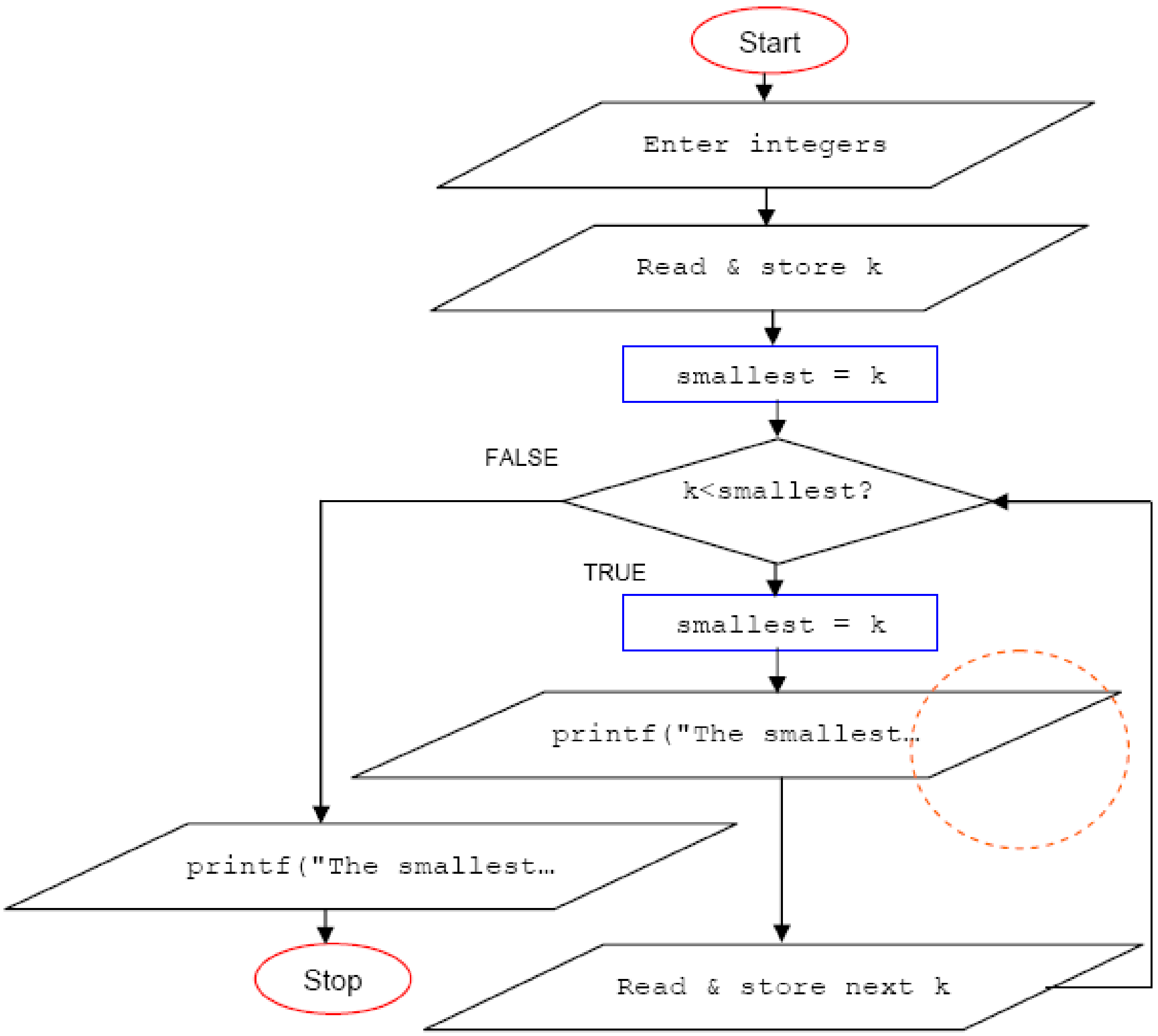
```
Enter integers, when done enter a negative number
11 22 13 19 16 -1
The smallest number is 11 and it is data item #1
Press any key to continue . . .
```

To see the program flow clearer, you can add some codes as shown below and the output sample is on the right side.

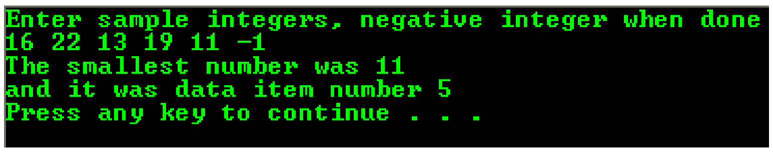
```
#include <stdio.h>

int main(void)
{
    int k, smallest;
    printf("Enter integers, when");
    printf(" done enter a ");
    printf("negative number\n");
    scanf_s("%d", &k, 1);
    // assign the first number to
smallest variable
    smallest = k;
    printf("smallest=%d, k=%d pos1\n",
smallest, k);
    // iterate while k >= 0
    for( ; k >= 0; )
    {
        // if the entered number is <
smallest
        if(k < smallest)
        {
          // then assign the number to
smallest variable...
          smallest = k;
          // do some checking...
          printf("smallest=%d, k=%d pos2
\n", smallest, k);
          printf("The smallest number
has just been changed to %d\n",
smallest);
        }
        // read the next input....repeat
        scanf_s("%d", &k, 1);
        printf("smallest=%d, k=%d pos3
\n", smallest, k);
    }
    // print the smallest number...
    printf("The smallest number is %d
\n", smallest);
    return 0;
}
```



```
Enter integers, when done enter a negative number
16
smallest=16, k=16 pos1
22
smallest=16, k=22 pos3
13
smallest=16, k=13 pos3
smallest=13, k=13 pos2
The smallest number has just been changed to 13
19
smallest=13, k=19 pos3
11
smallest=13, k=11 pos3
smallest=11, k=11 pos2
The smallest number has just been changed to 11
-1
smallest=11, k=-1 pos3
The smallest number is 11
Press any key to continue . . .
```



15. Enter a sample input: 16, 22, 13, 19, 11, -1 for the following experiment. The program will determine the smallest number entered.


```
Enter sample integers, negative integer when done
16 22 13 19 11 -1
The smallest number was 11
and it was data item number 5
Press any key to continue . . .
```

```
#include <stdio.h>

int main(void)
{
    int i, k, smallest, item_num;
    printf("Enter sample integers, negative
integer when done\n");
    scanf_s("%d", &k, 1);
    // assign the first input to variable
smallest
    smallest = k;
    // initialize the item_num to 1
    item_num = 1;
    // start the for loop
    for(i = 1; k >= 0; i = i + 1)
    {
        // if the entered number is <
smallest...
        if(k < smallest)
        {
            // assign the entered number to
smallest
            smallest = k;
            // assign the count to
item_num...
            item_num = i;
```

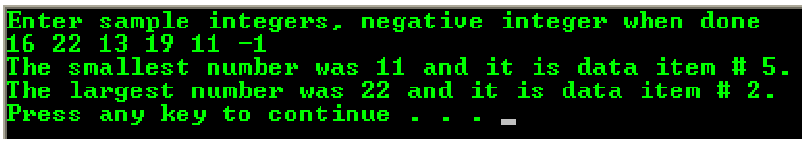
a. This is left for your assignment!

```
#include <stdio.h>

int main(void)
{
  int i, k, smallest, item_num = 1, item_num1 = 1, largest;
  printf("Enter sample integers, negative integer when done\n");
  scanf_s("%d", &k, 1);
  // assign the first input to variable smallest & largest
  smallest = k;
  largest = k;
  // start the for loop
  for(i = 1; k >= 0; i = i + 1)
  {
    // if the entered number is < smallest...
    if(k < smallest)
    {
      // assign the entered number to smallest
      smallest = k;
      // assign the count to item_num...
      item_num = i;
    }
   // if the entered number is > largest...
```

```
        }
        // read next input...repeat
        scanf_s("%d", &k, 1);
    }
    // print the result...
   printf("The smallest number was %d\n",
smallest);
   printf("and it was data item number %d
\n", item_num);
   return 0;
}
```

a. Draw a tracechart and note how i and item_num change.
b. Try altering the program so that it prints the smallest and the largest numbers at the end of the loop. You may disregard the variable item_num. However, you will now need a variable called largest.

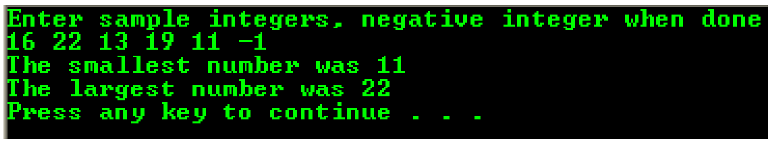
```
    if(k > largest)
    {
      // assign the entered number to largest
      largest = k;
      // assign the count to item_num1...
      item_num1 = i;
    }
    // read next input...repeat
    scanf_s("%d", &k, 1);
  }
  // print the result...
  printf("The smallest number was %d and it is data item # %d.
\n", smallest, item_num);
  printf("The largest number was %d and it is data item # %d.
\n", largest, item_num1);
  return 0;
}
```



```
Enter sample integers, negative integer when done
16 22 13 19 11 -1
The smallest number was 11 and it is data item # 5.
The largest number was 22 and it is data item # 2.
Press any key to continue . . .
```

16. Use a sample input: 16, 22, 13, 19, 11 and -1 for the following experiment. This program will determine the smallest and the largest numbers entered.

```
#include <stdio.h>

int main(void)
{
    int k, smallest, largest;
    printf("Enter sample integers, negative
integer when done\n");
    scanf_s("%d", &k, 1);
    // assign the first input to variable
smallest and largest
    smallest = k;
    largest = k;
    // start the for loop
    for( ; k >= 0; )
    {
        // if the entered number is <
smallest...
        if(k < smallest)
            // assign the entered number to
smallest
            smallest = k;
        // if the entered number is >
largest...
        if(k > largest)
            // assign the entered number to
largest
            largest = k;
        // read next input...repeat
        scanf_s("%d", &k, 1);
    }
    // print the result...
    printf("The smallest number was %d\n",
smallest);
    printf("The largest number was %d\n",
largest);
    return 0;
}
```

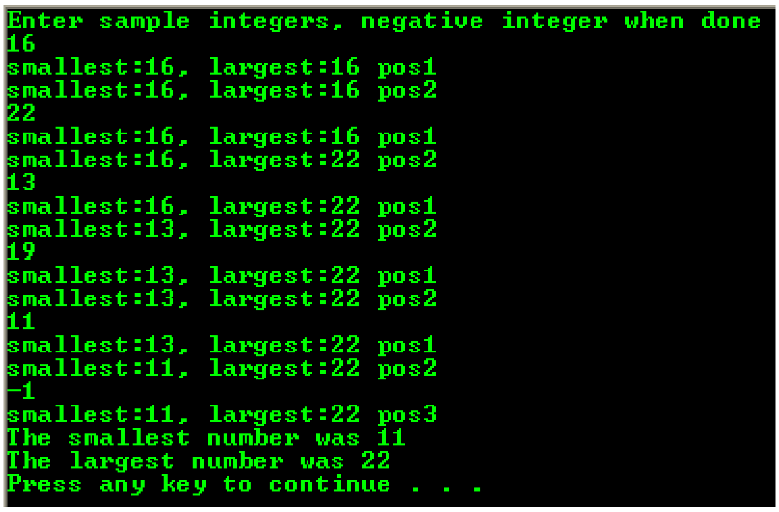


```
Enter sample integers, negative integer when done
16 22 13 19 11 -1
The smallest number was 11
The largest number was 22
Press any key to continue . . .
```

b. The largest is 16, 13 and 11 and the smallest is 16 and 22. You can see this by adding some codes as shown below.

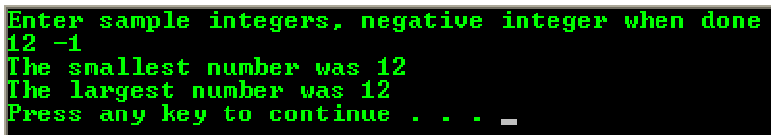
```
#include <stdio.h>

int main(void)
{
  int k, smallest, largest;
  printf("Enter sample integers, negative integer when done\n");
  scanf_s("%d", &k, 1);
  // assign the first input to variable smallest and largest
  smallest = k;
  largest = k;
  // start the for loop
  for( ; k >= 0; )
  {
    printf("smallest:%d, largest:%d pos1\n", smallest, largest);
    // if the entered number is < smallest...
    if(k < smallest)
    // assign the entered number to smallest
    smallest = k;
    if(k > largest)
    largest = k;
    // read next input...repeat
    printf("smallest:%d, largest:%d pos2\n", smallest,    largest);
    scanf_s("%d", &k, 1);
}
// print the result...
printf("smallest:%d, largest:%d pos3\n", smallest, largest);
printf("The smallest number was %d\n", smallest);
printf("The largest number was %d\n", largest);
return 0;
}
```

a. Draw the flowchart.
b. What were the different values of the smallest and the largest variables?
c. Is it possible for both if statements to be true for a given k?
d. When using the following data: 16, 12, 17, 19, 11 and -1, what were the different values of smallest? What were the different values of largest?



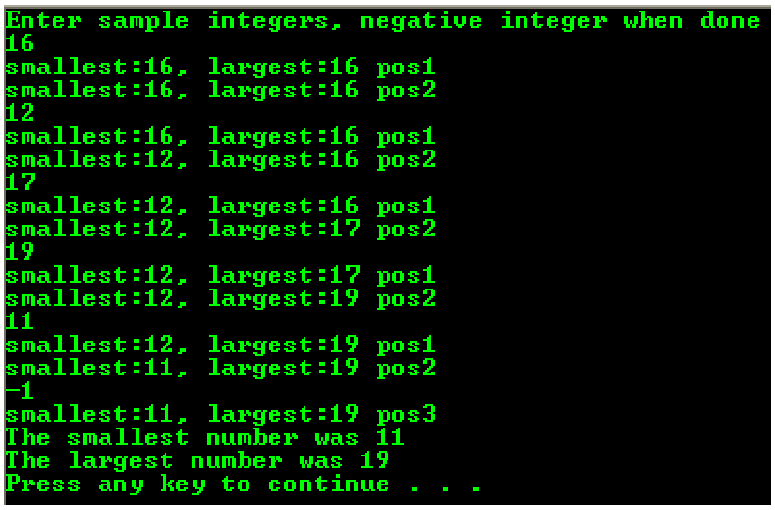
```
Enter sample integers, negative integer when done
16
smallest:16, largest:16 pos1
smallest:16, largest:16 pos2
22
smallest:16, largest:16 pos1
smallest:16, largest:22 pos2
13
smallest:16, largest:22 pos1
smallest:13, largest:22 pos2
19
smallest:13, largest:22 pos1
smallest:13, largest:22 pos2
11
smallest:13, largest:22 pos1
smallest:11, largest:22 pos2
-1
smallest:11, largest:22 pos3
The smallest number was 11
The largest number was 22
Press any key to continue . . .
```

c. Yes, possible. This happens when only one input has been entered.



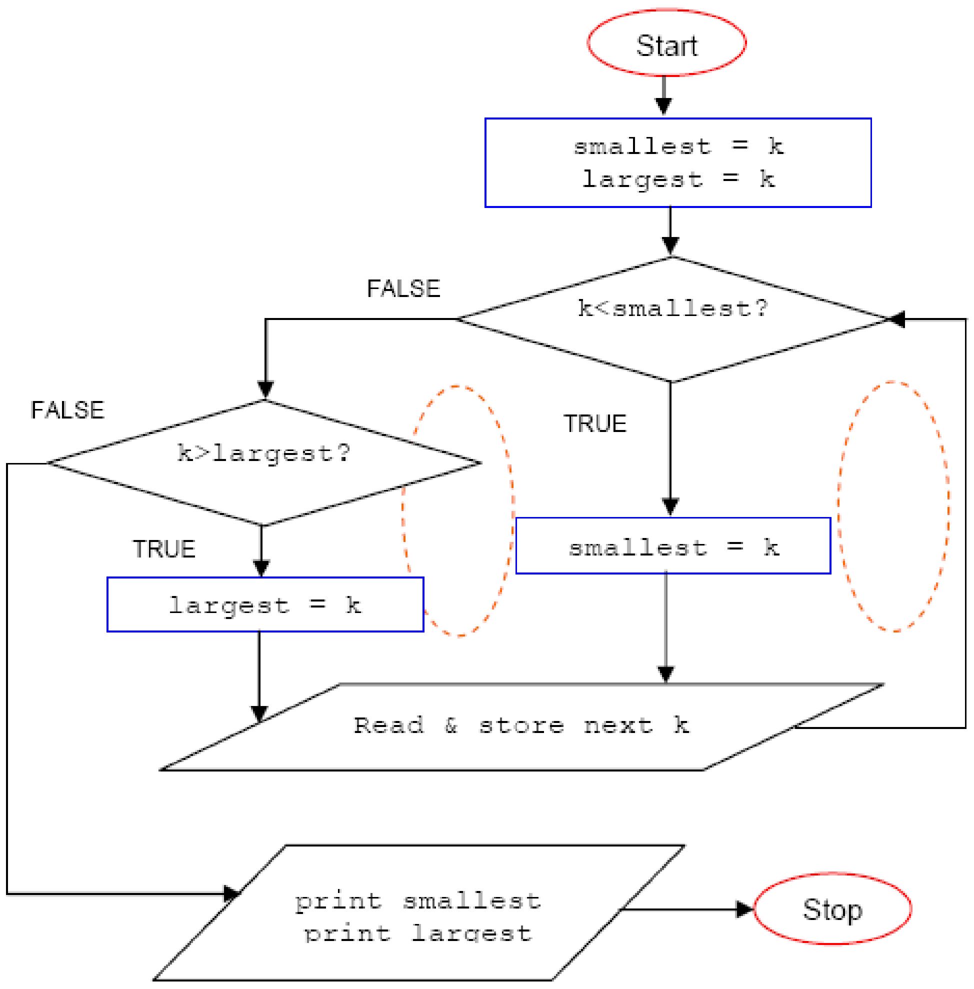
```
Enter sample integers, negative integer when done
12 -1
The smallest number was 12
The largest number was 12
Press any key to continue . . .
```

d. Different values for smallest are 16, 12 and 11. Different values for largest are 16, 17 and 19.



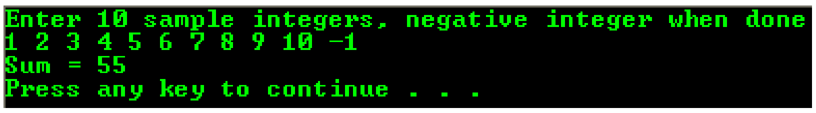
```
Enter sample integers, negative integer when done
16
smallest:16, largest:16 pos1
smallest:16, largest:16 pos2
12
smallest:16, largest:16 pos1
smallest:12, largest:16 pos2
17
smallest:12, largest:16 pos1
smallest:12, largest:17 pos2
19
smallest:12, largest:17 pos1
smallest:12, largest:19 pos2
11
smallest:12, largest:19 pos1
smallest:11, largest:19 pos2
-1
smallest:11, largest:19 pos3
The smallest number was 11
The largest number was 19
Press any key to continue . . .
```

- The following is the flowchart diagram for the previous question.

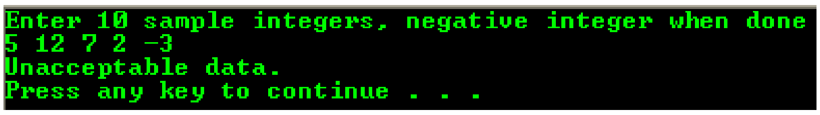


17. Next, let try using a break command. Run the following program twice. For the first run, use 10 positive integers, for the second, use 5, 12, 7, 2 and -3.

```
#include <stdio.h>

int main(void)
{
    int i, k, sum = 0, flag = 0;
    printf("Enter 10 sample integers,
negative integer when done\n");
    for(i = 1; i <= 10; i = i + 1)
    {
        scanf_s("%d", &k, 1);
        // if k < 0...
        if(k < 0)
        {
            flag = 1;
            // break the loop...
            break;
        }
        // for k > 0, do the sum and repeat...
        sum = sum + k;
    }
    if(flag == 1)
        printf("Unacceptable data.\n");
    else
        printf("Sum = %d\n", sum);
    return 0;
}
```



```
Enter 10 sample integers, negative integer when done
1 2 3 4 5 6 7 8 9 10 -1
Sum = 55
Press any key to continue . . .
```



```
Enter 10 sample integers, negative integer when done
5 12 7 2 -3
Unacceptable data.
Press any key to continue . . .
```

a. 10 times based on the for terminal condition, i <=10.
b. No. The loop terminates.
c. Skip or break the current processing.
d. To skip processing when certain condition is met.
e. No.

a. When all positive numbers (k >

0) entered, how many times did the loop executed?

b. When a negative number was entered, did the loop execute the same number of times?

c. What do you think the break statement does?

d. When do you think it may come in handy?

e. The for and if statements use parentheses. Functions such as printf() and strcpy() or strcpy_s() also use parentheses. Does break use parentheses?

- From the previous experiments, we already explored how to combine two decisions into one, how to negate or reverse a logical expression and how to terminate a loop in midstream if statement, for example, if an abnormal condition has occurred.
- For C/C++ programs it is very important to indent the code properly. This will provide readability and maintainability. Fortunately, newer compilers provide automatic indentation. In the exercises that you have done, only a simple C codes used and what about if the codes span hundreds or thousand line of codes?
- Try using the same relational operator throughout a flowchart for consistency. For example, use only the >= operator in a given flowchart because you can achieve the same result.
- Keep the true on the right side and false on the left side of decision diamonds whenever possible. These rules make the logic easy to follow.

18. For the following codes, indent correctly and draw the flowchart.

a. if(sex == 'm') if (age > 50) mold = mold + 1; else myoung = myoung + 1;
else if (age > 50) fold = fold + 1;
else fyoung = fyoung + 1;

```
if(sex == 'm')
    if (age > 50)
        mold = mold + 1;
    else
        myoung = myoung + 1;
else if (age > 50)
    fold = fold + 1;
else fyoung = fyoung + 1;
```

b. if (age > 50) { if (smokes == 'y') if (weight > 150)
risk_factor = 10;
else risk_factor = 2; if
(excercises == 'n') risk_factor =
risk_factor + 2; } else printf
("Forget about it!\n");

```
if (age > 50)
{
    if (smokes == 'y')
        if (weight > 150)
            risk_factor = 10;
        else risk_factor = 2;
        if (excercises == 'n')
            risk_factor = risk_factor + 2;
}
else printf("Forget about it!\n");
```

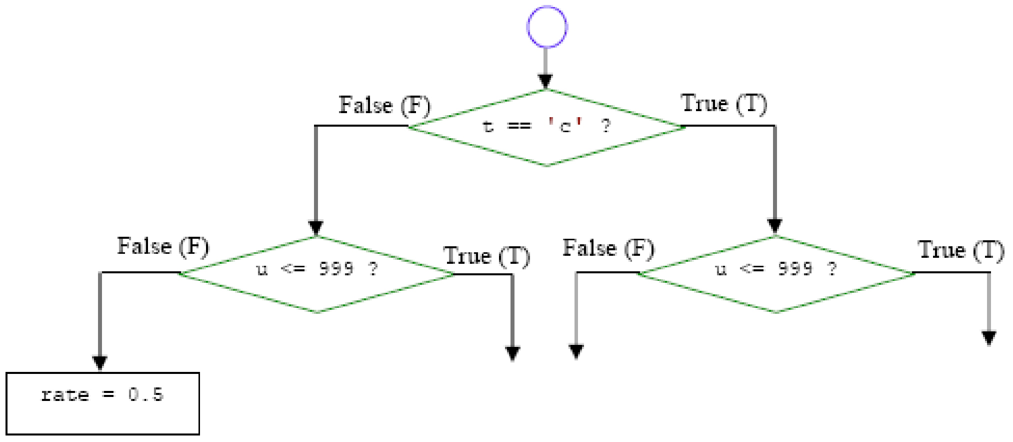
19. Using the provided data in the Table, complete the following flowchart to assign a proper value of rate and then write a full working program.

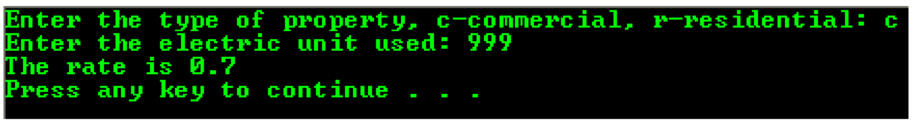
Unit used (u)	Residential rate	Commercial rate
0 < units <= 200	0.8	0.6
200 < units <= 999	0.7	0.3
Above 900	0.5	0.2
code	'r'	'c'

Table 2

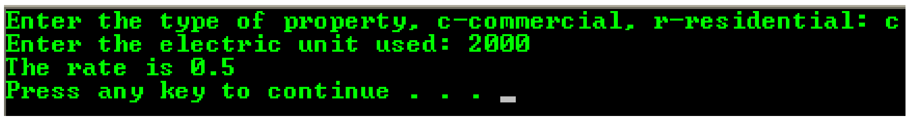
```
#include <stdio.h>

int main(void)
{
    double r = 0.0;
    int electric_unit_used = 0;
    char prop_type = '1';
    printf("Enter the type of property, c-commercial, r-
residential: ");
    scanf_s("%c", &prop_type, sizeof(char));
    printf("Enter the electric unit used: ");
    scanf_s("%d", &electric_unit_used, sizeof(int));
    if(prop_type == 'c')
    {
        if(electric_unit_used <= 200)
        {
          r = 0.8;
          printf("The rate is %.1f\n", r);
        }
        else if (electric_unit_used <= 999)
        {
           r = 0.7;
           printf("The rate is %.1f\n", r);
        }
        else
        {
          r = 0.5;
          printf("The rate is %.1f\n", r);
        }
    }
    if(prop_type == 'r')
    {
        if(electric_unit_used <= 200)
        {
           r = 0.6;
           printf("The rate is %.1f\n", r);
        }
       else if (electric_unit_used <= 999)
       {
           r = 0.3;
           printf("The rate is %.1f\n", r);
       }
       else
       {
          r = 0.2;
          printf("The rate is %.1f\n", r);
       }
    }
    return 0;
}
```

- Partial flowchart example for the previous question.

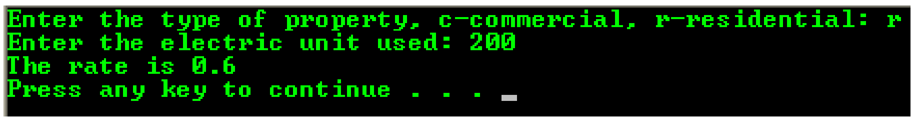




```
Enter the type of property, c-commercial, r-residential: c
Enter the electric unit used: 999
The rate is 0.7
Press any key to continue . . .
```

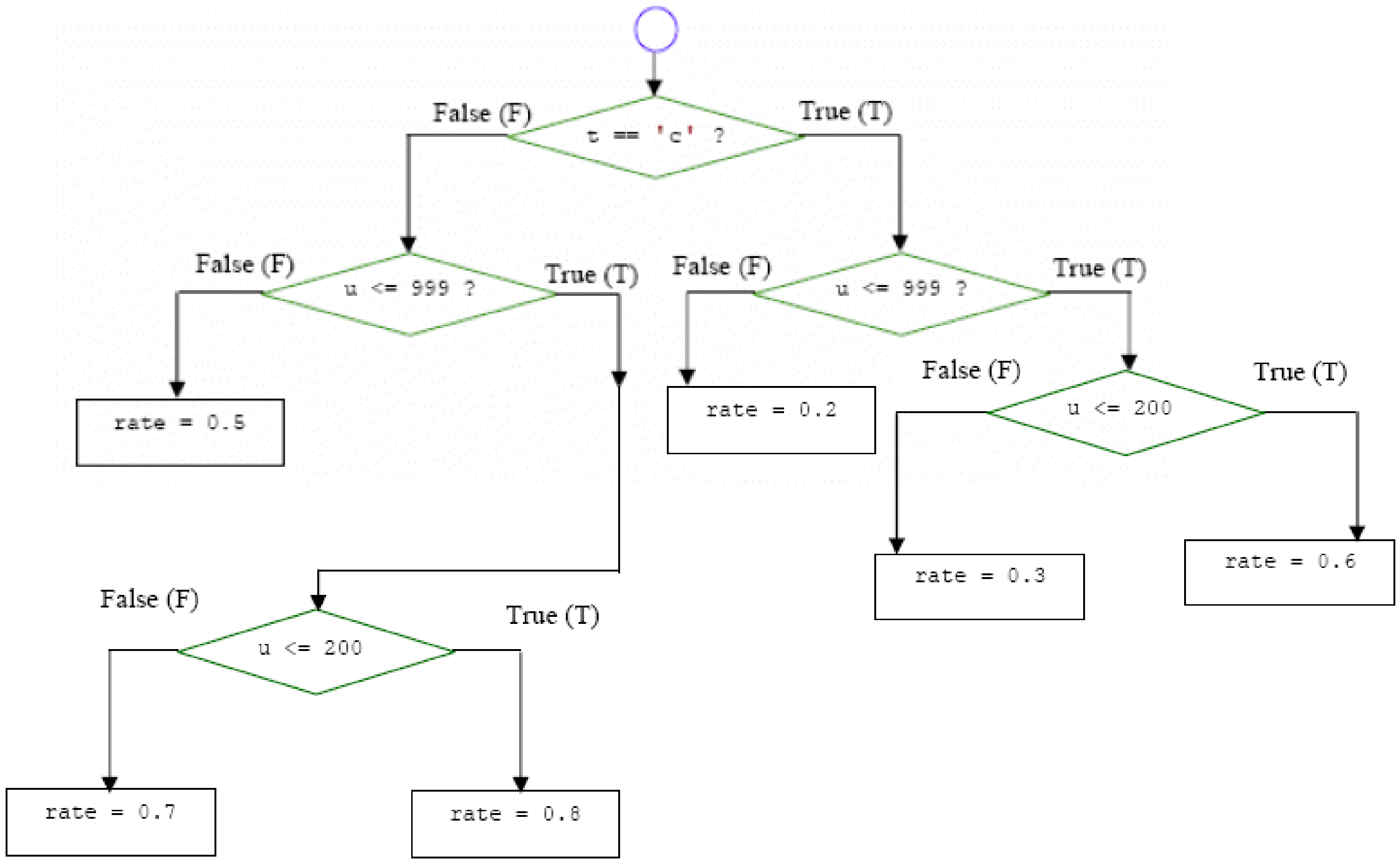


```
Enter the type of property, c-commercial, r-residential: c
Enter the electric unit used: 2000
The rate is 0.5
Press any key to continue . . . _
```



```
Enter the type of property, c-commercial, r-residential: r
Enter the electric unit used: 200
The rate is 0.6
Press any key to continue . . . _
```

- The following is a completed flowchart diagram for the previous question.



The C Selection if, if-else, if-else-if, break, conditional/ternary operator and switch-case-break: Part 1 | Part 2 | Part 3 | Part 4 | Part 5 | Part 6