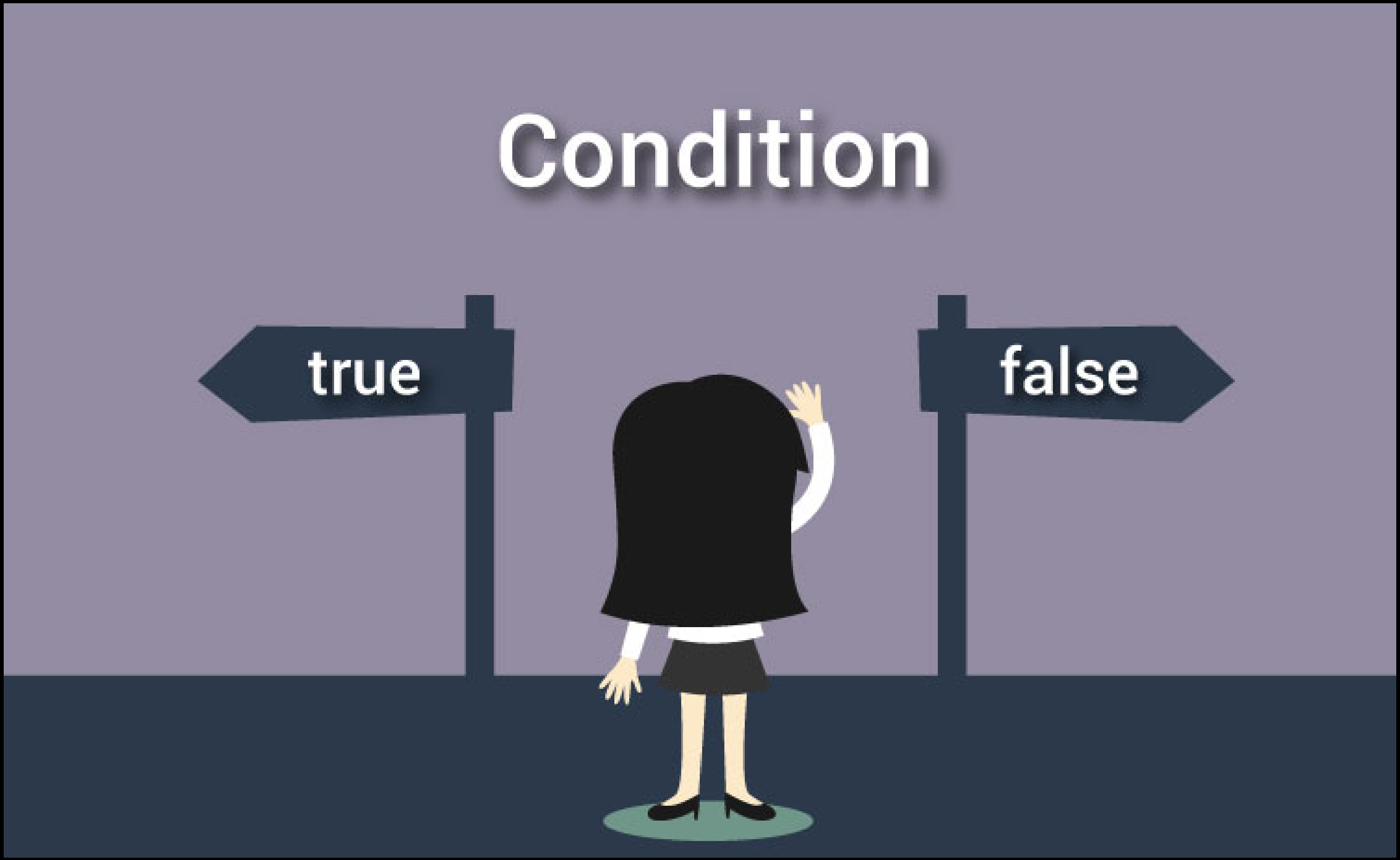
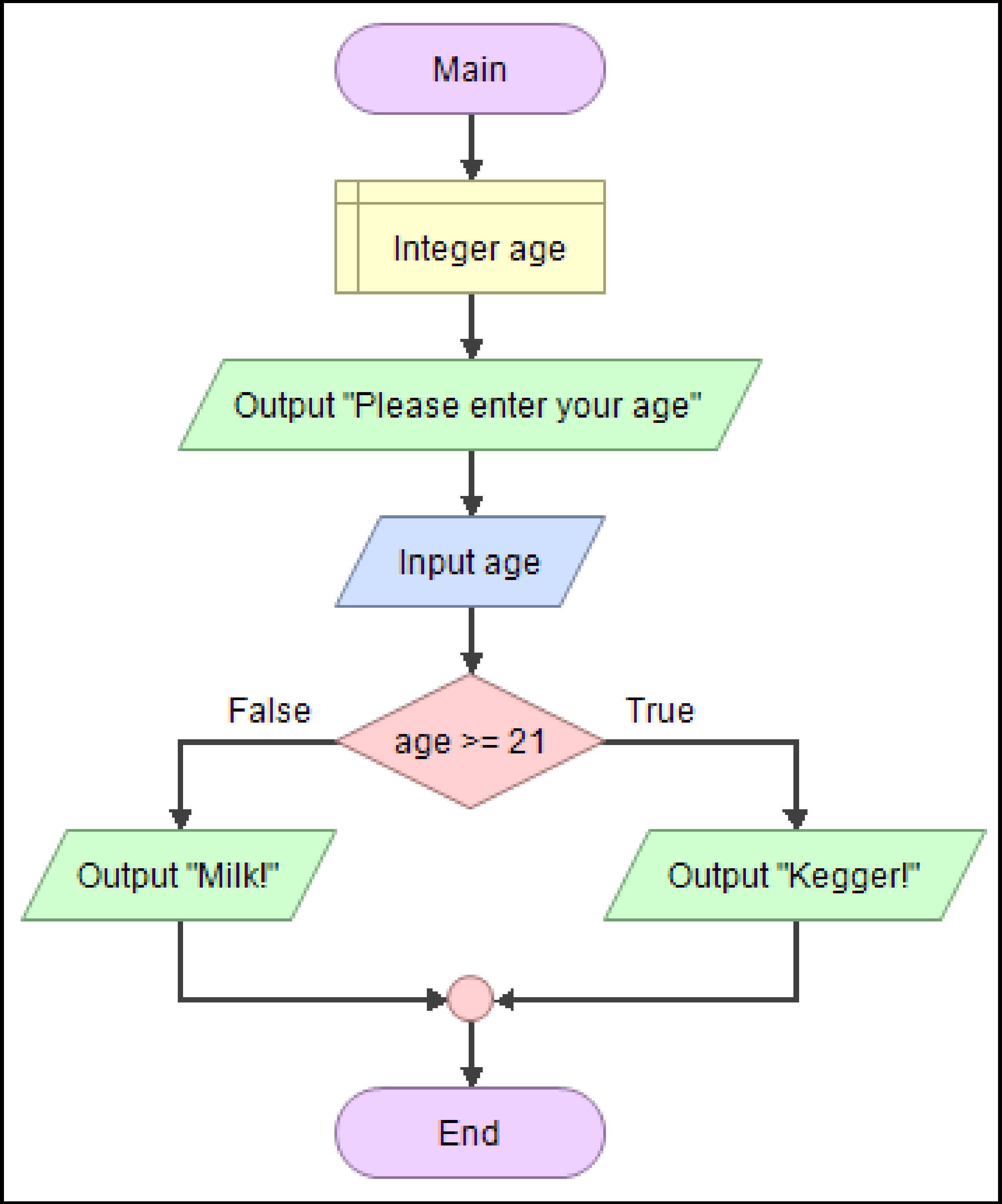
CSE102
Computer Programming

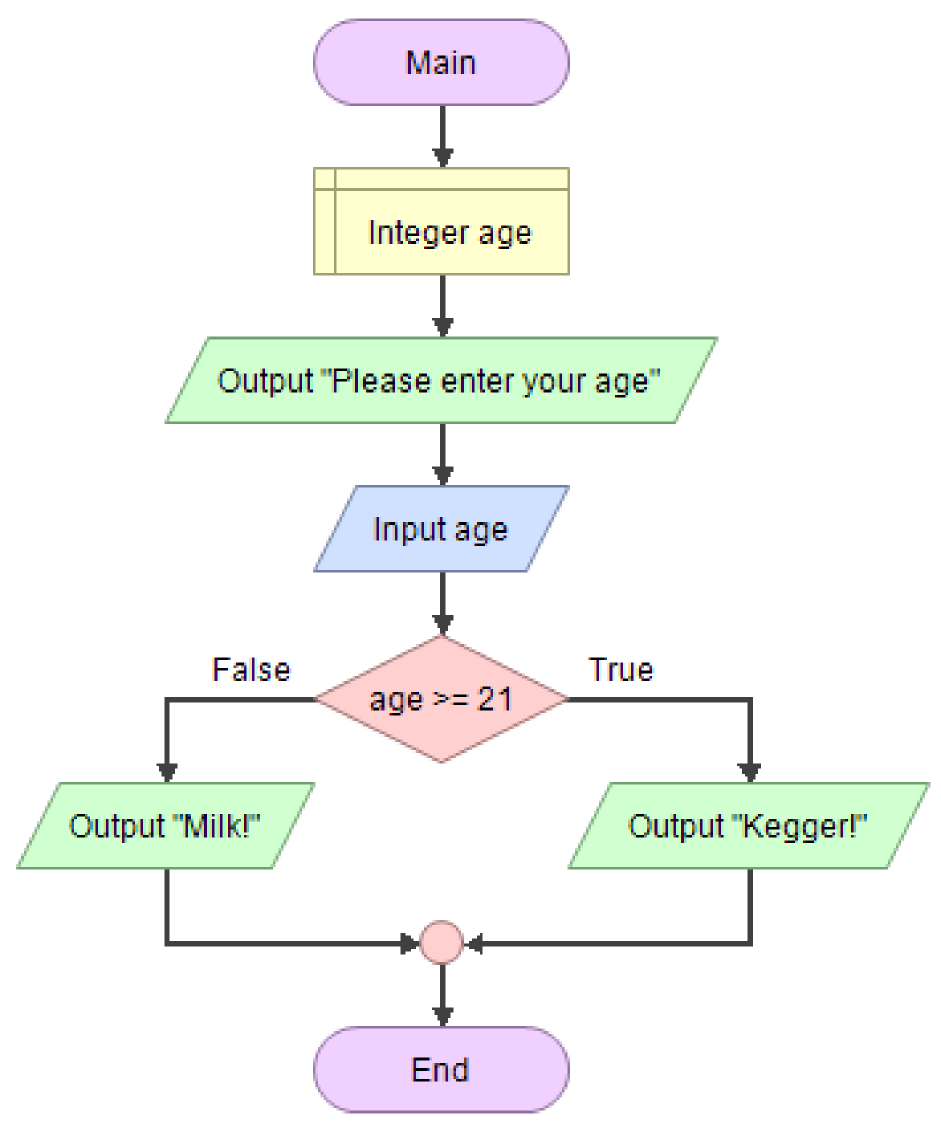


Credits: programiz.com

Predictable yet Unknown

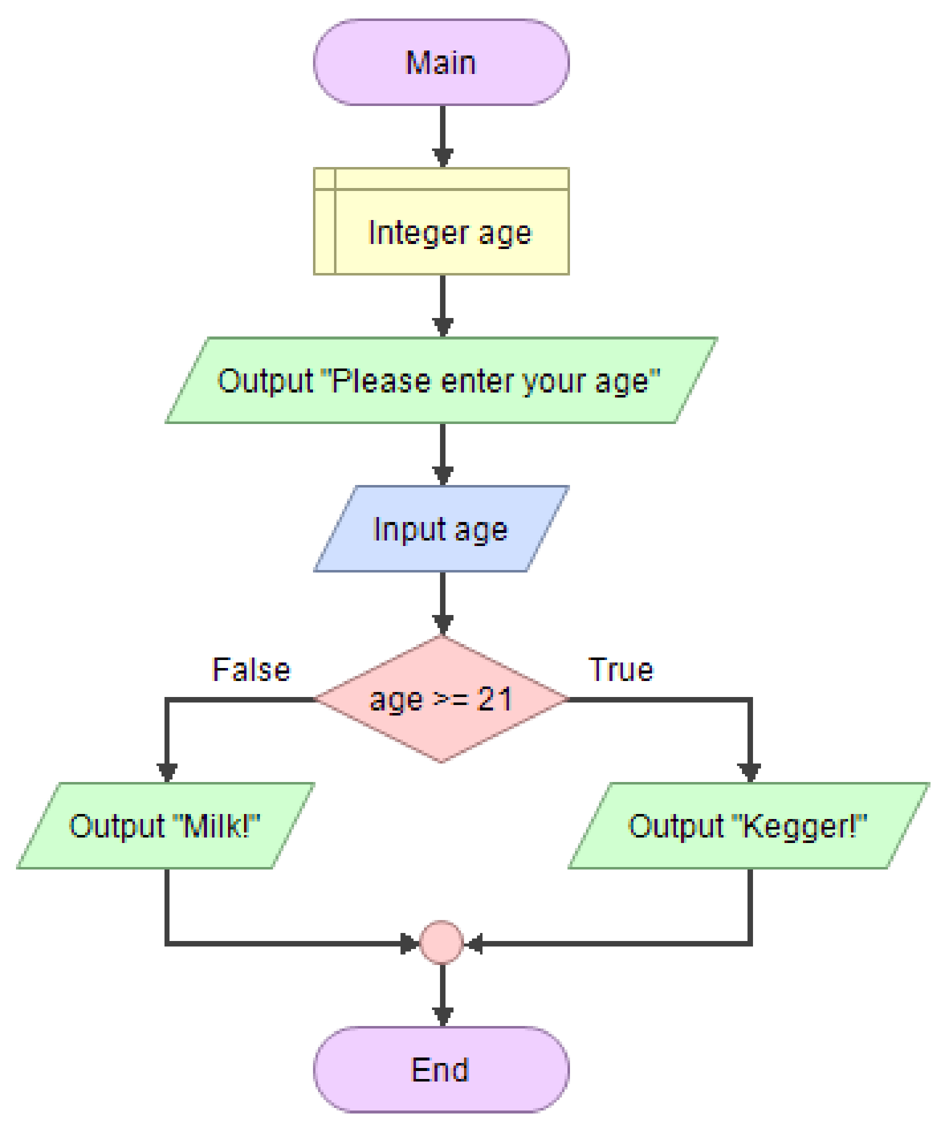


Selection



```
int main()
{



    return 0;
}
```

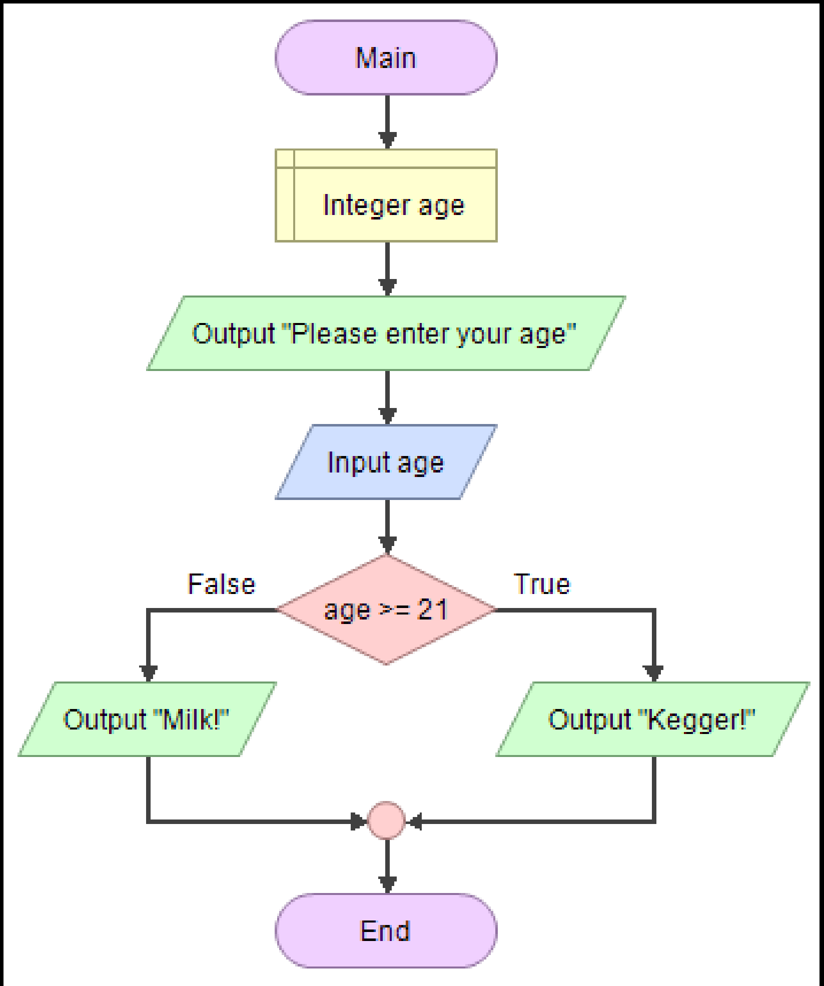
Selection



```
int main()
{
  int age;

  return 0;
}
```

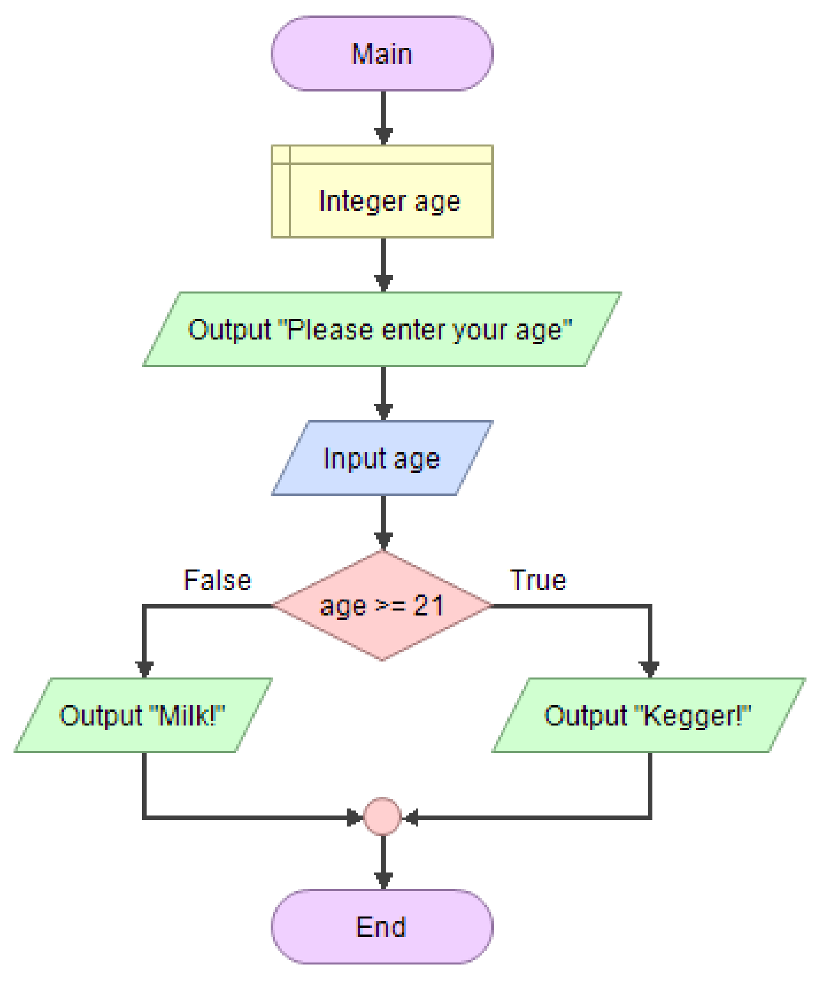
Selection

```
#include <stdio.h>
int main()
{
  int age;
  printf(“Please enter
             your age ”);

  return 0;
}
```

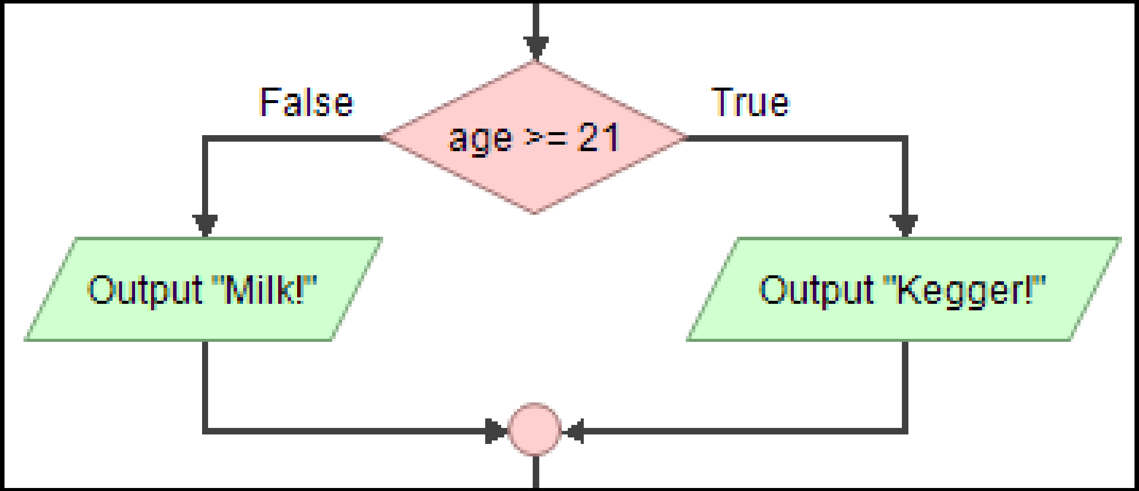


Selection

```
#include <stdio.h>
int main()
{
  int age;
  printf(“Please enter
              your age ”);
  scanf(“%d”, &age);

  return 0;
}
```

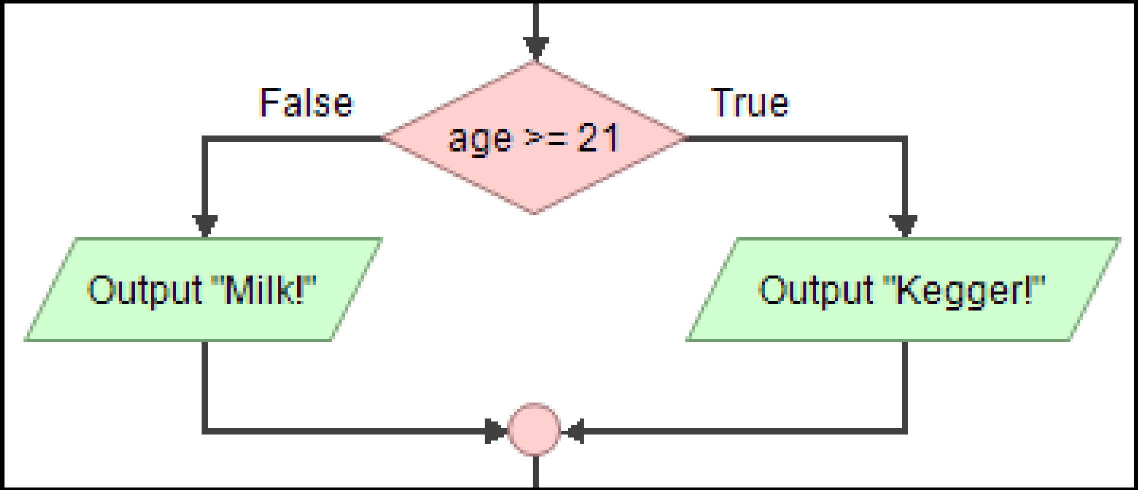


Selection



```
if (age >= 21){
  printf(“Kegger!\n);
}
else{
  printf(“Milk!\n”);
}
```

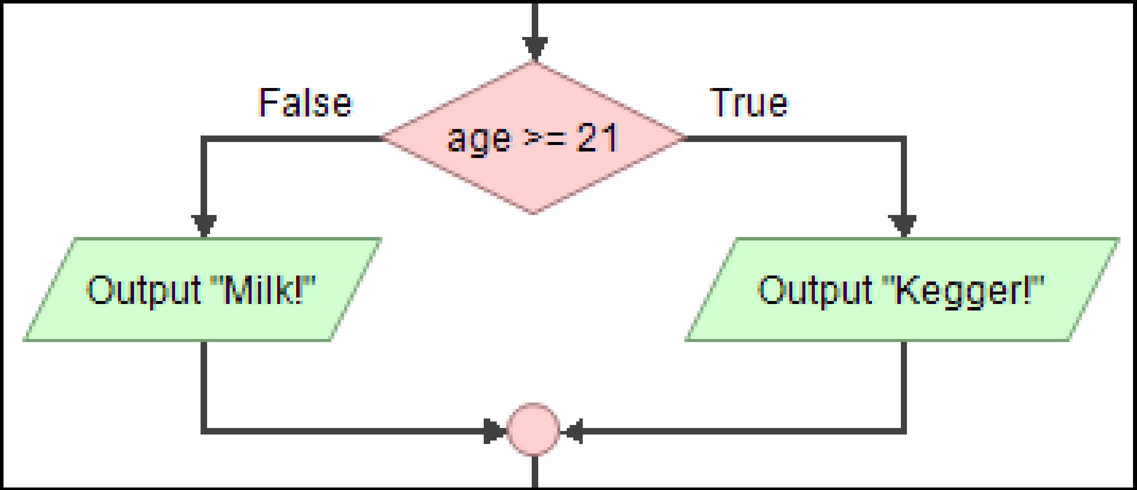
Selection



If age is greater than or equal to 21

```
if (age >= 21){
  printf(“Kegger!\n);
}
else{
  printf(“Milk!\n”);
}
```

Selection

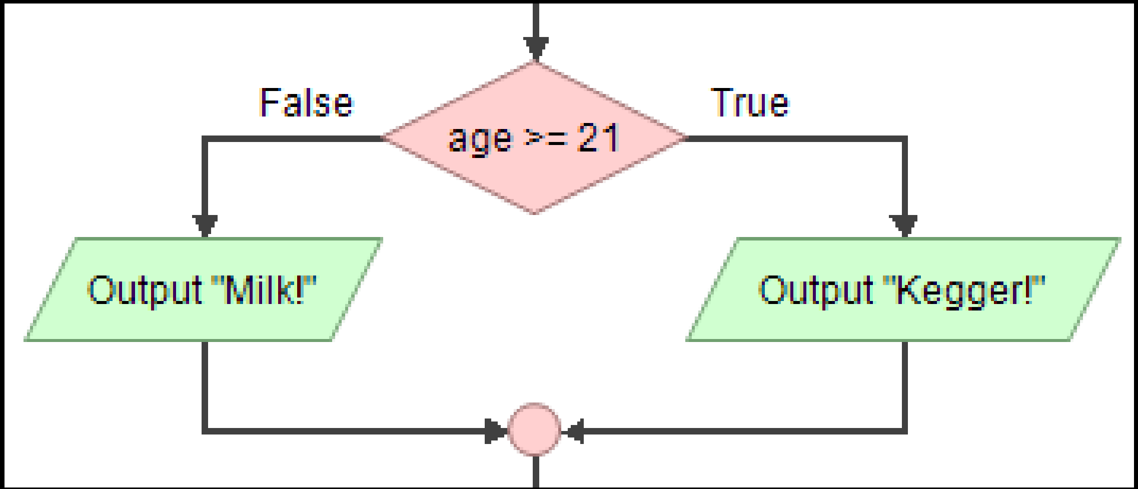


If age is greater than or equal to 21

then print kegger

```
if (age >= 21){
   printf(“Kegger!\n);
}
else{
   printf(“Milk!\n”);
}
```

Selection

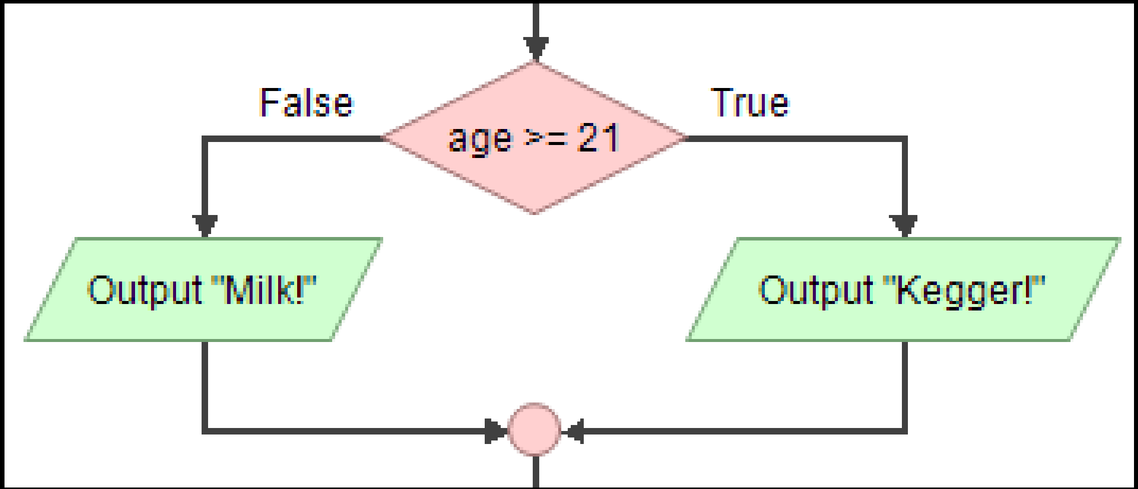


If ~~age is greater than or equal to 21~~ ~~then print kegger~~

```
if (age >= 21){
  printf(“Kegger!\n);
}
else{
  printf(“Milk!\n”);
}
```

else i.e age less than 21

Selection



If ~~age is greater than or equal to 21~~

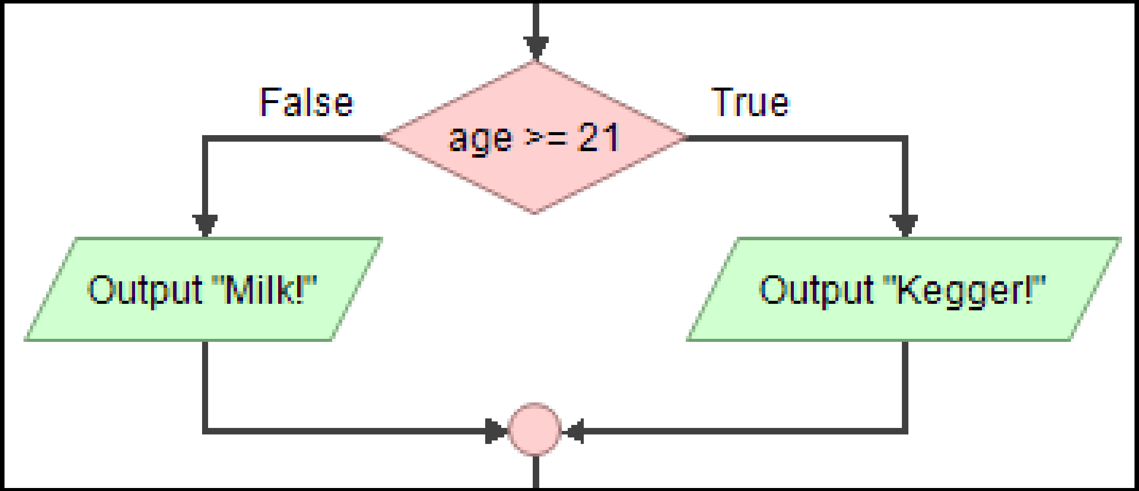
~~then print kegger~~

```
if (age >= 21){
  printf(“Kegger!\n);
}
else{
  printf(“Milk!\n”);
}
```

else i.e age less than 21

print milk

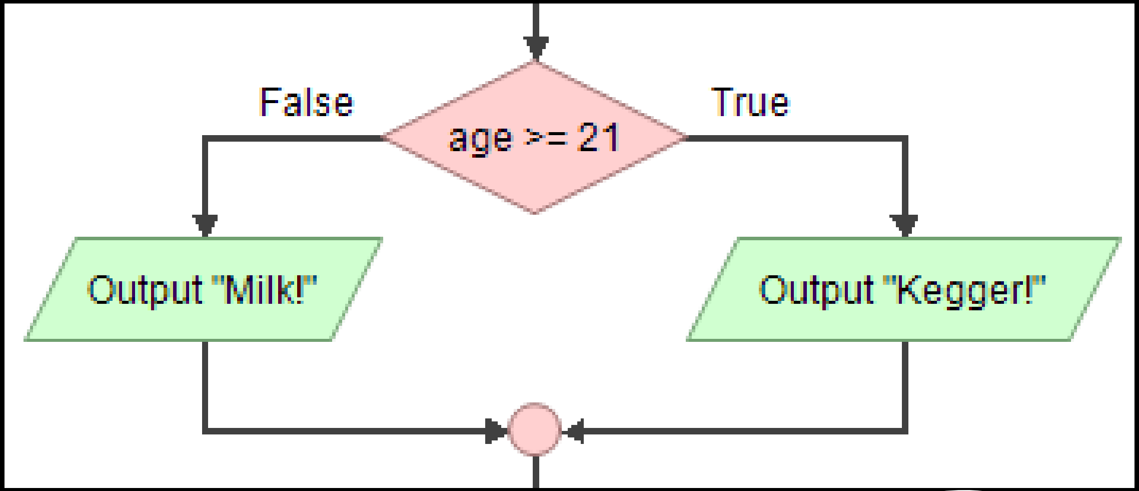
Selection



```
if (age >= 21){
   printf(“Kegger!\n);
}
else{
   printf(“Milk!\n”);
}
```

If keyword - in literal stands for questioning the truth of test expression

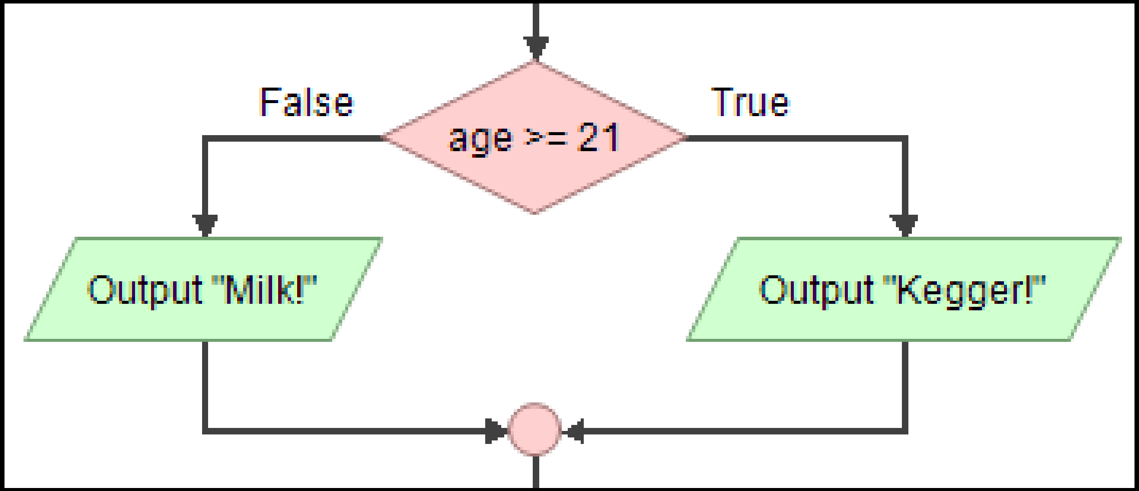
Selection



```
if (age >= 21){
  printf(“Kegger!\n);
}
else{
  printf(“Milk!\n”);
}
```

Test expression

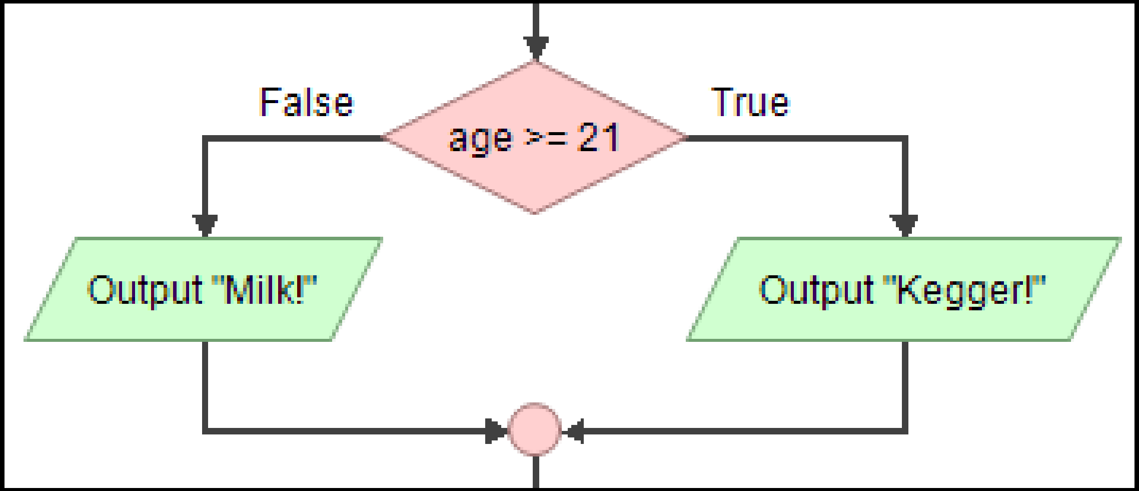
Selection



```
if (age >= 21){
   printf(“Kegger!\n);
}
else{
   printf(“Milk!\n”);
}
```

True part i.e
if expression is true
this statement will be
executed.
Note parantheses.

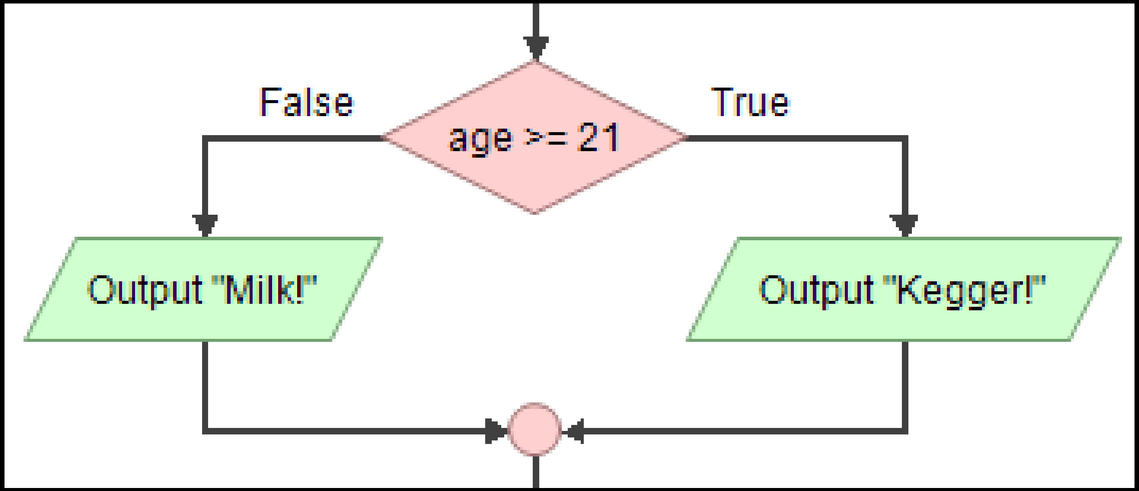
Selection



```
if (age >= 21){
  printf(“Kegger!\n);
}
else{
  printf(“Milk!\n”);
}
```

False part i.e
if expression is false
this statement will be
executed.

Selection



```
if (age >= 21){
  printf(“Kegger!\n);
}
else{
  printf(“Milk!\n”);
}
```

else keyword – in literal stands for failing of test expression

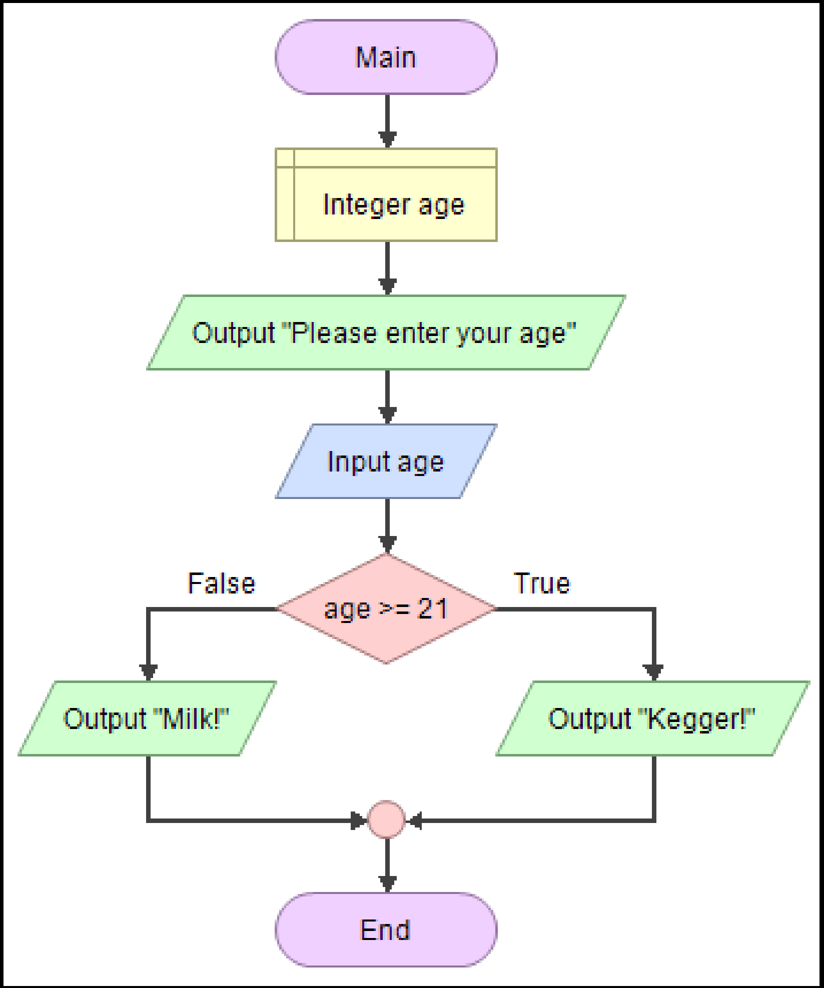
`if` Template

```
if (      ){

}
else{

}
```

if Template

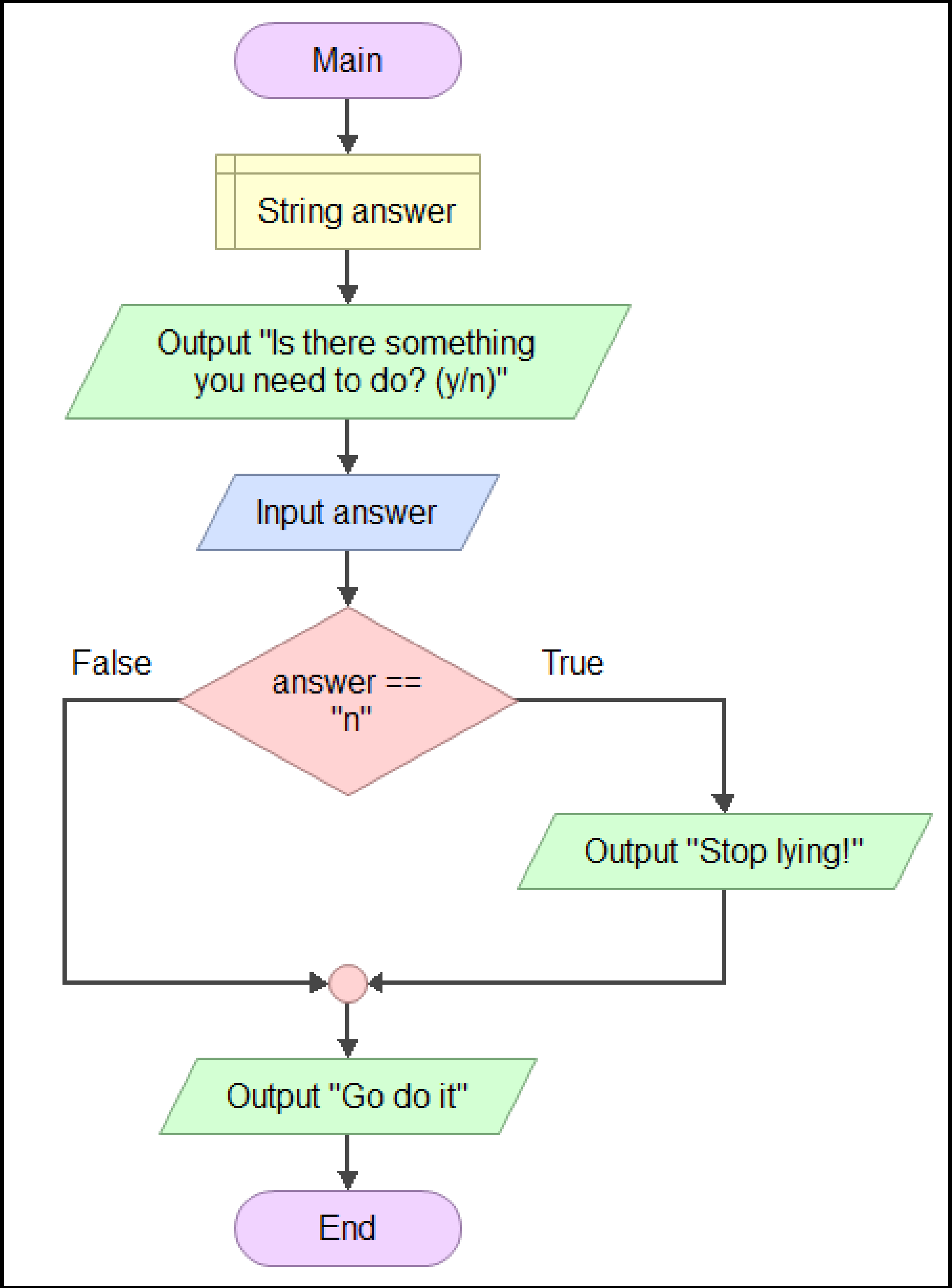
```
if (test-expression){
  //statement(s) to be executed
  //if test-expression is true
}
else{
  //statement(s) to be executed
  //if test-expression is false
}
```

Selection

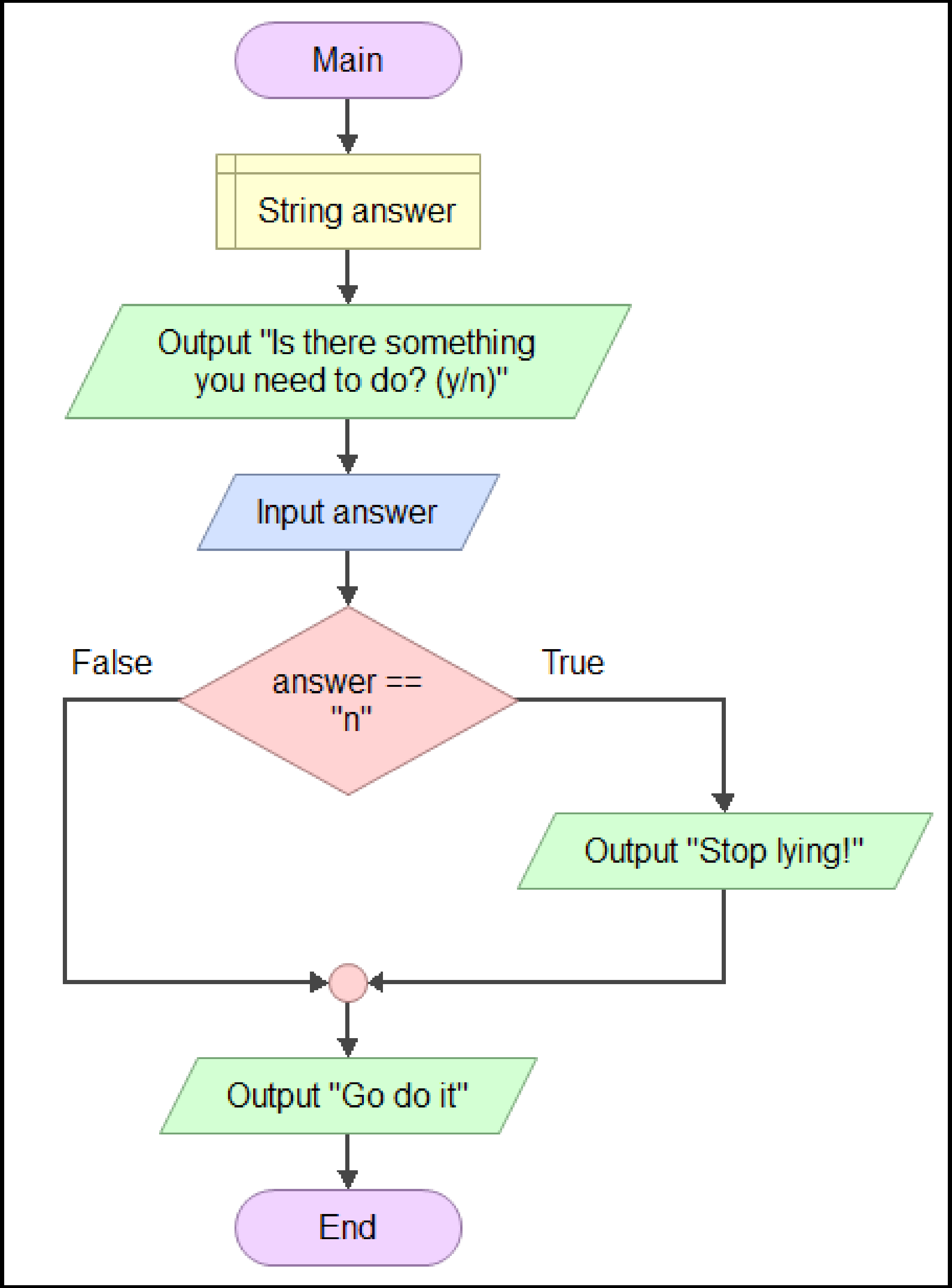


```
#include <stdio.h>
int main()
{
  int age;
  printf(“Please enter
          your age ”);
  scanf(“%d”, &age);
  if(age>=21){
    printf(“Kegger!”);
  }
  else{
    printf(“Milk!”);
  }

  return 0;
}
```

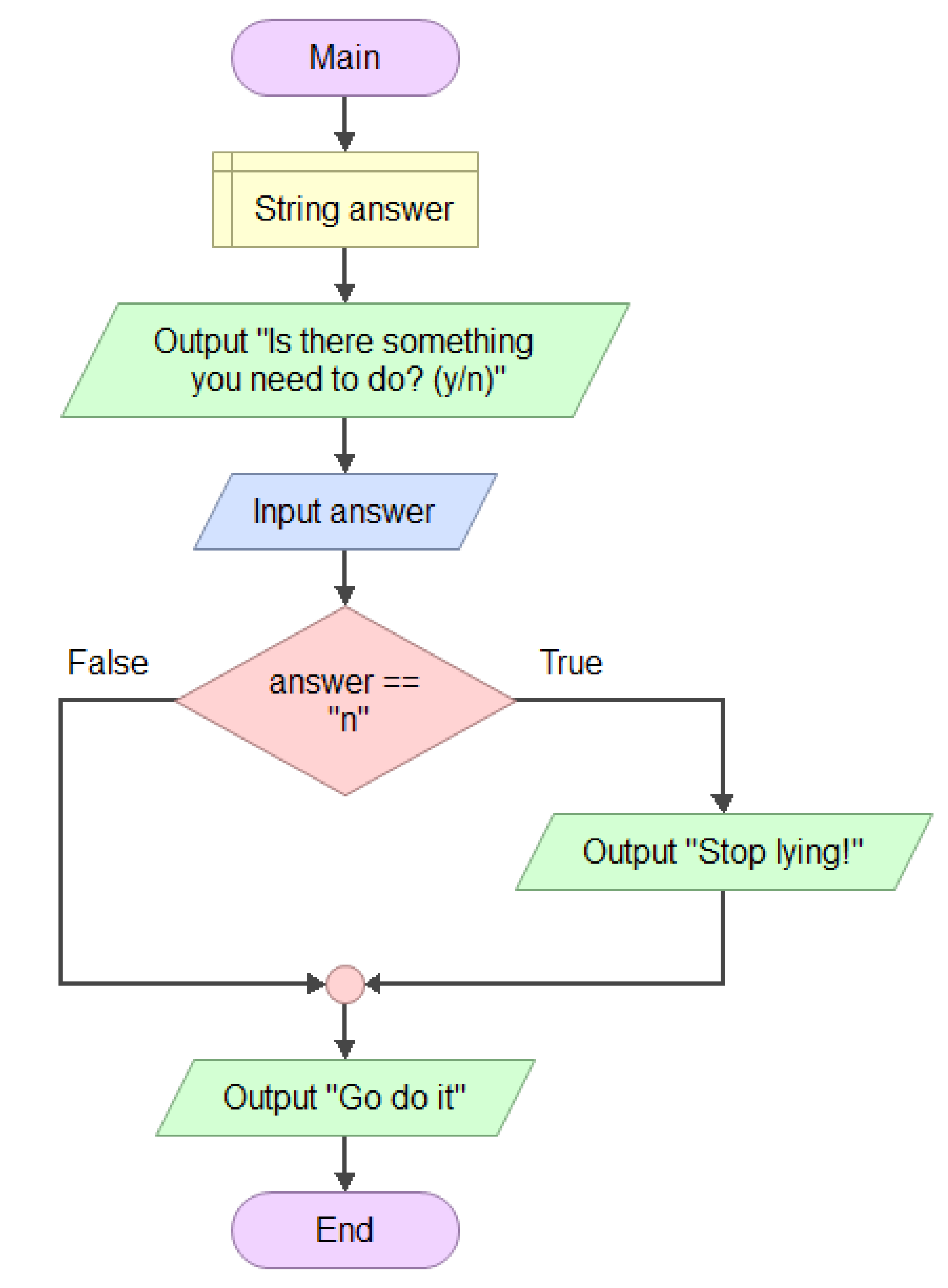
Another Example



Credits: flowgorithm.com



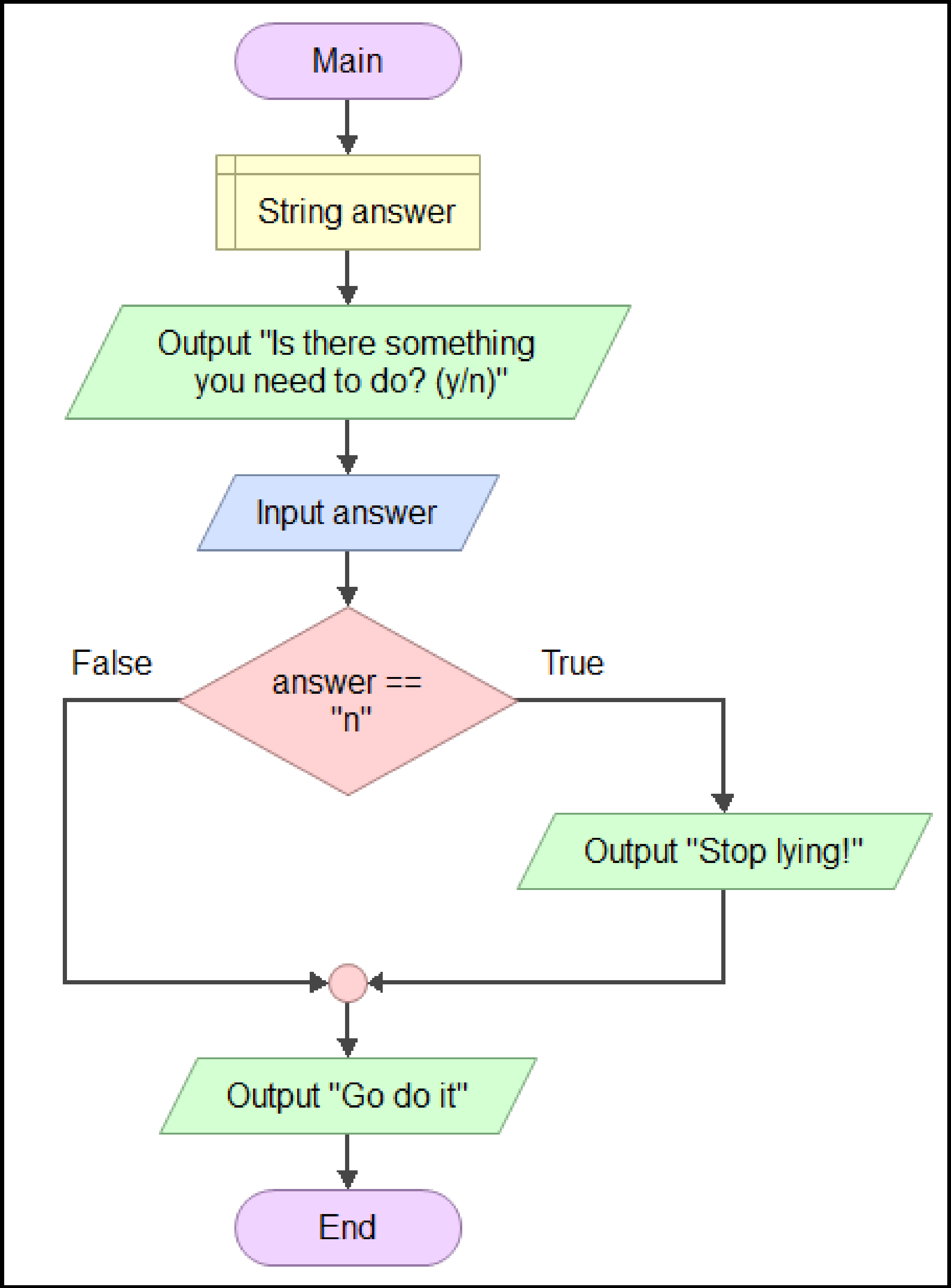
```
int main()
{



    return 0;
}
```



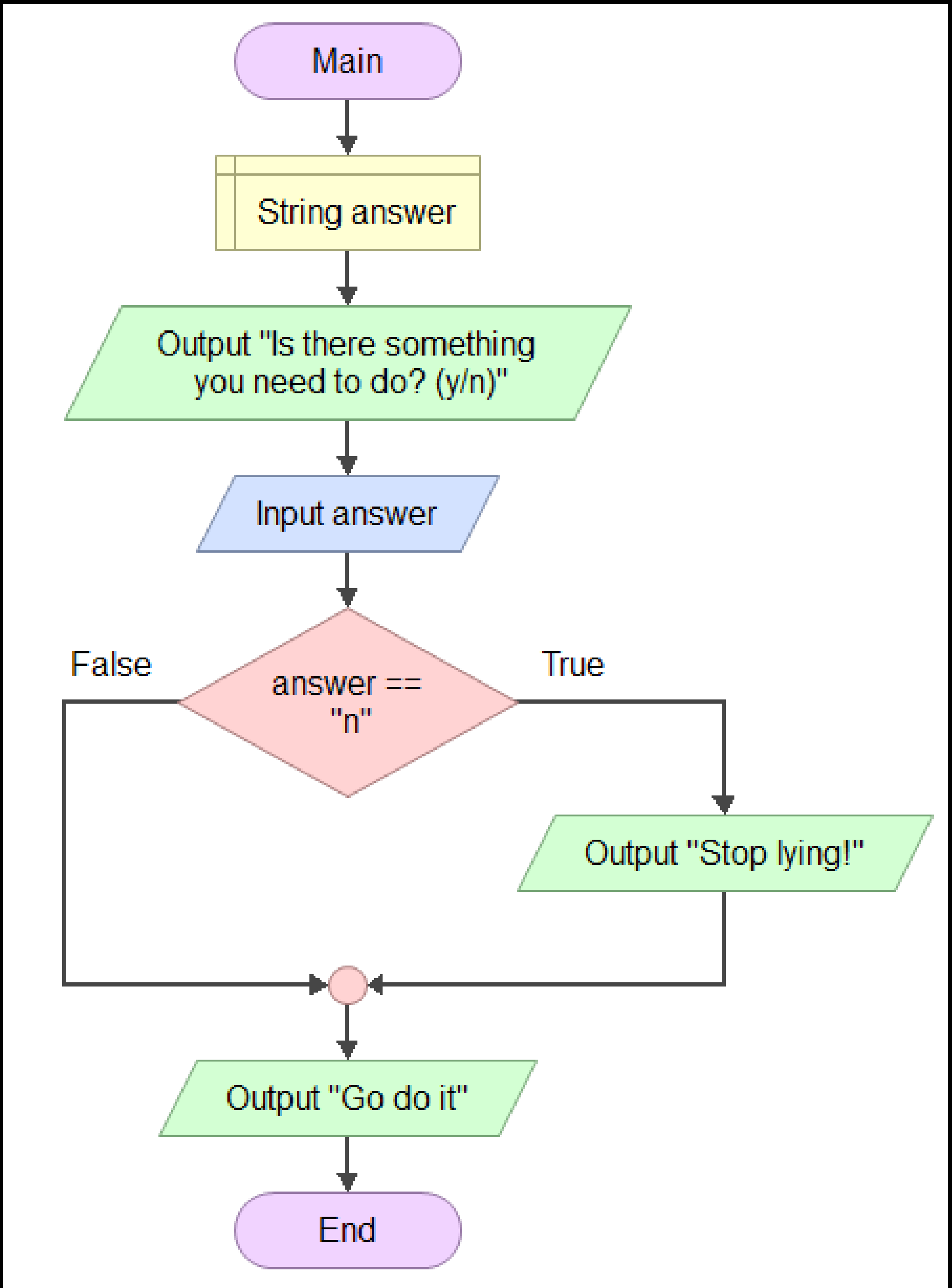
```
int main()
{
  char answer;

    return 0;
}
```

Character data type

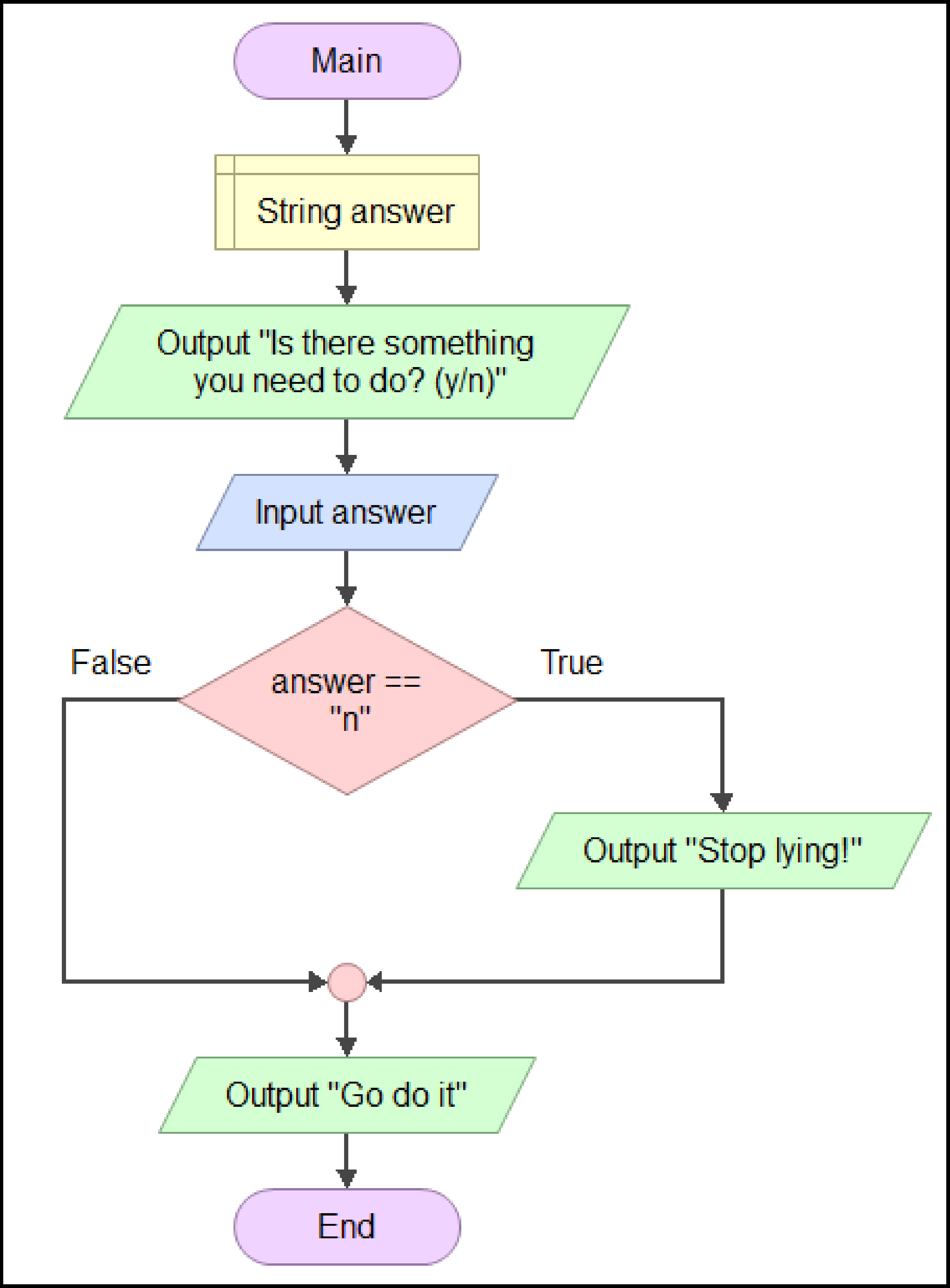
Strings are character arrays in C



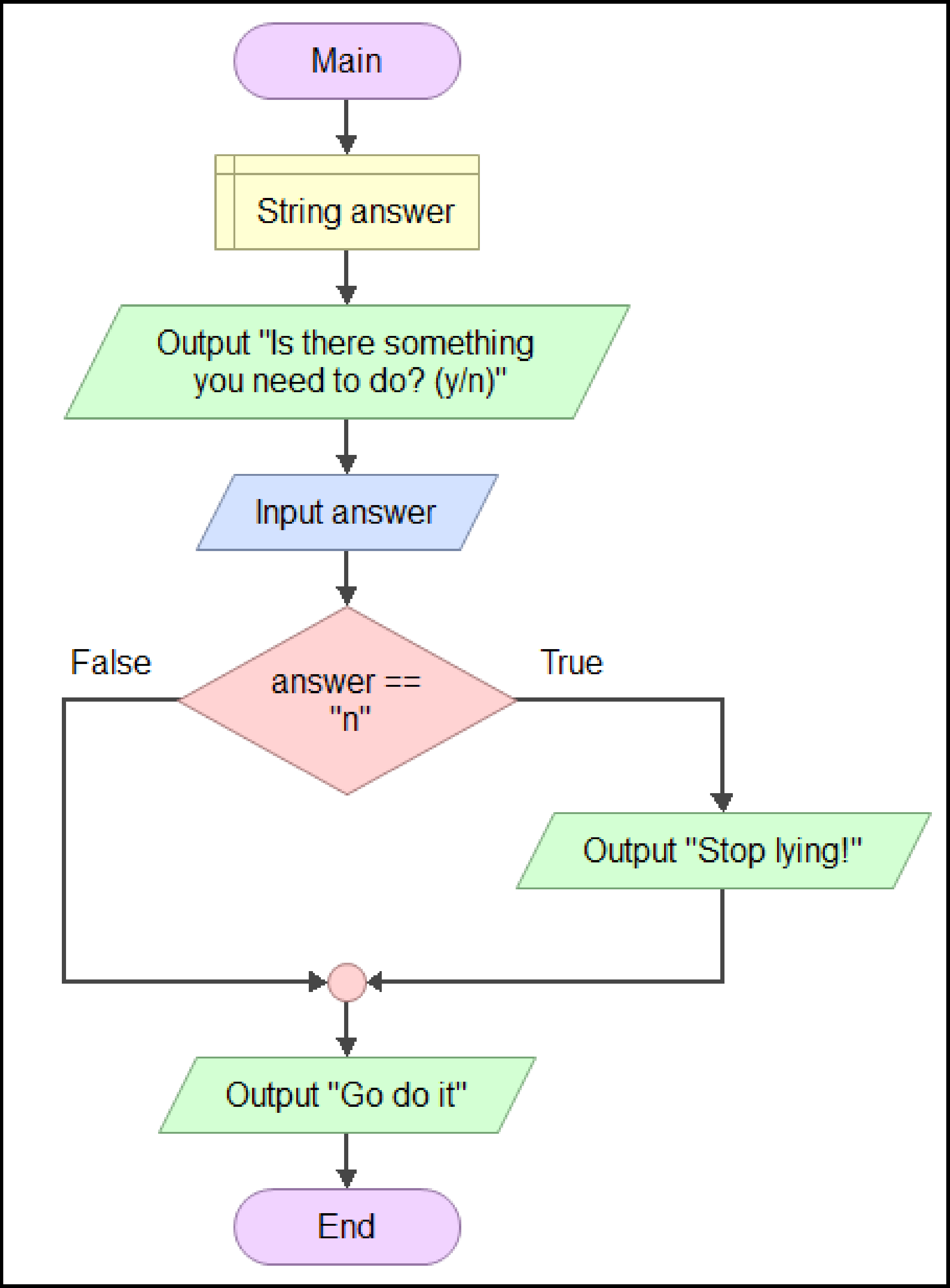
```
#include <stdio.h>
int main()
{
   char answer;
   printf(“Is there
    something you need
    to do? (y/n) ”);



      return 0;
}
```



```
#include <stdio.h>
int main()
{
    char answer;
    printf("Is there
      something you need
      to do? (y/n) ");
    scanf("%c", &answer);


        return 0;
}
```



```
#include <stdio.h>
int main()
{
    char answer;
    printf("Is there
     something you need
     to do? (y/n) ");
    scanf("%c", &answer);
    if(answer=='n'){
     printf("stop
               lying!\n");
    }
    printf("Go do it\n");
    return 0;
}
```

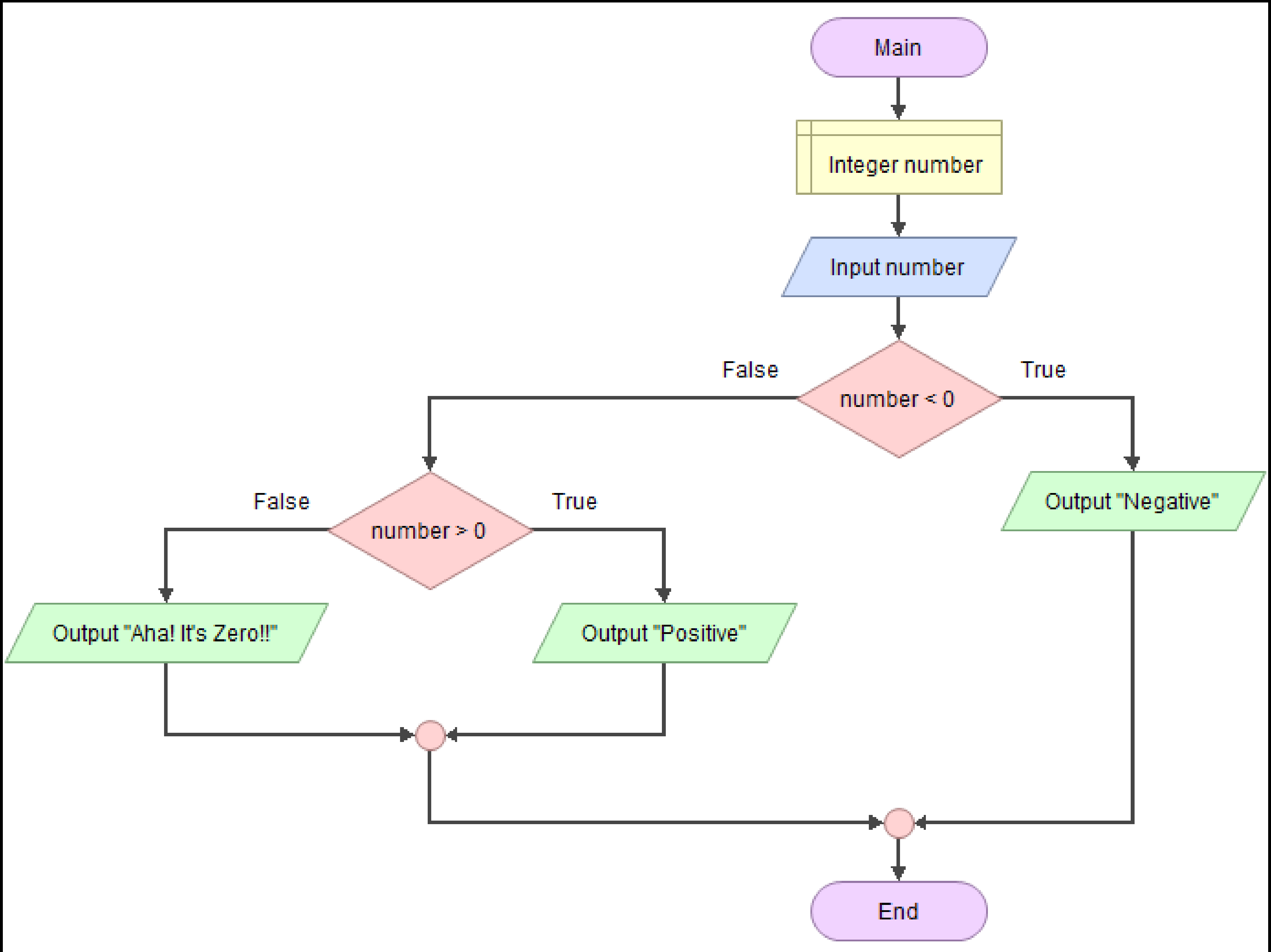


```
#include <stdio.h>
int main()
{
   char answer;
   printf(“Is there
    something you need
    to do? (y/n) ”);
   scanf(“%c”, &answer);
   if(answer==’n’){
    printf(“stop
              lying!\n”);
   }
   printf(“Go do it\n”);
   return 0;
}
```

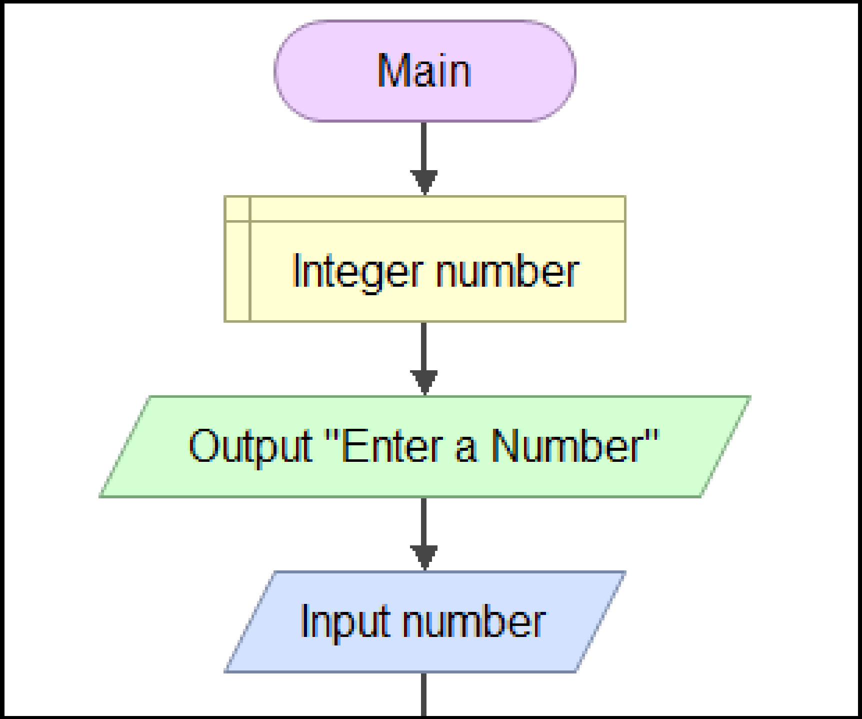
" " is for strings

' ' is for character

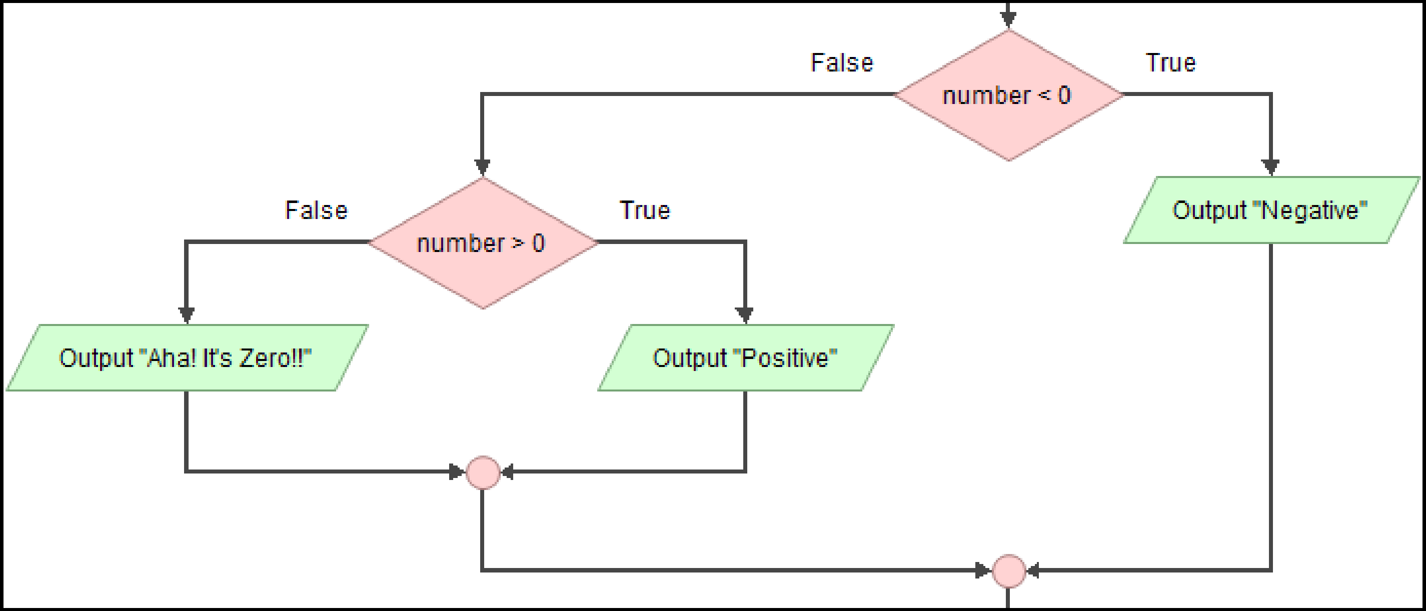
Nested Selection Example



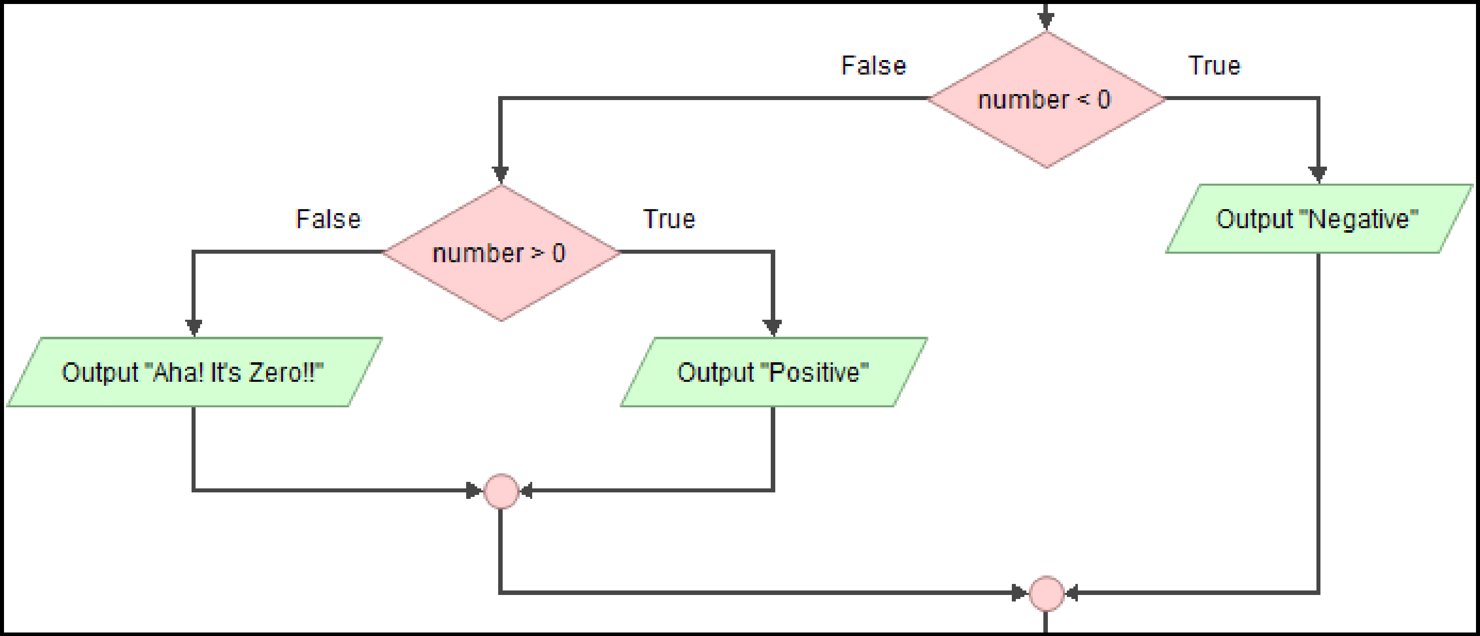
Credits: flowgorithm.com



```
#include <stdio.h>
int main()
{
  int number;
  printf(“Enter a
                number ”);
  scanf(“%d”, &number);

  return 0;
}
```

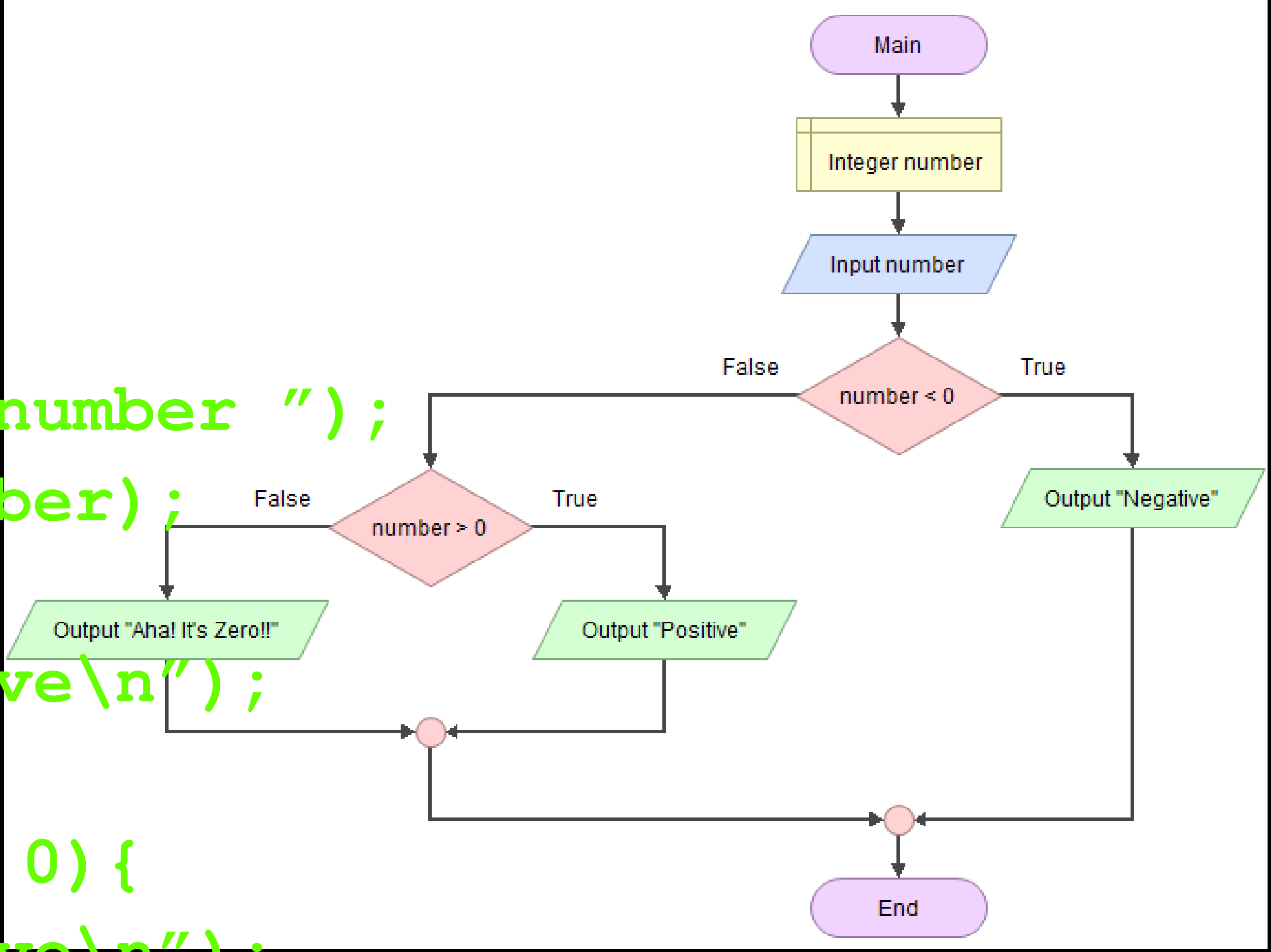


```
if(number < 0){
  printf(“Negative\n”);
}
else{
  if(number > 0){
    printf(“Positive\n”);
  }
  else{
    printf(“Aha! It’s Zero!!\n);
  }
}
```

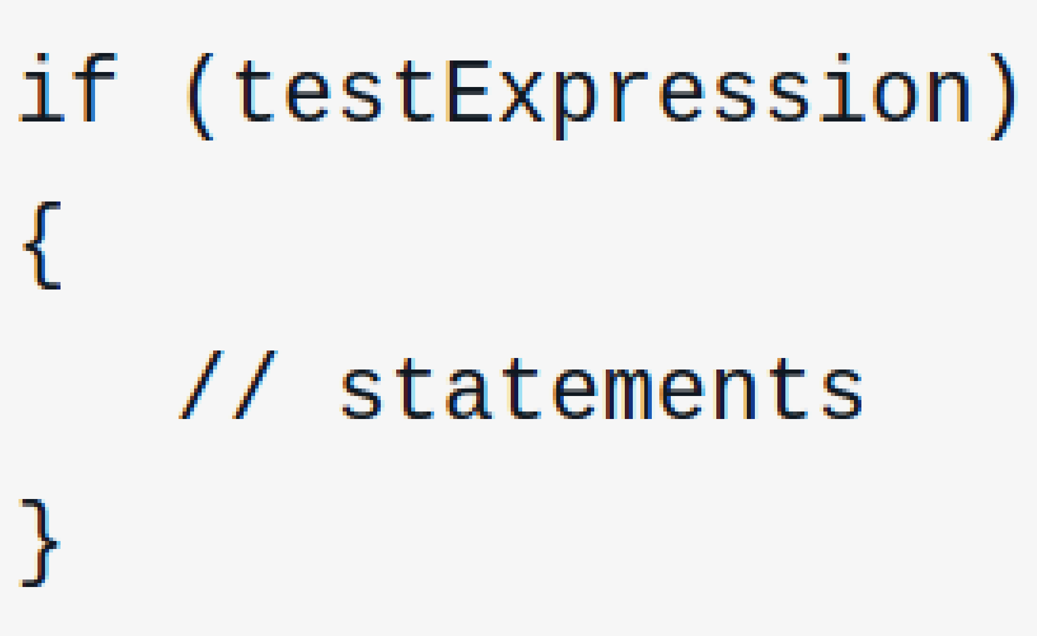


```
if(number < 0){
  printf(“Negative\n”);
}
else if(number > 0){
  printf(“Positive\n”);
}
else{
  printf(“Aha! It’s Zero!!\n);
}
```

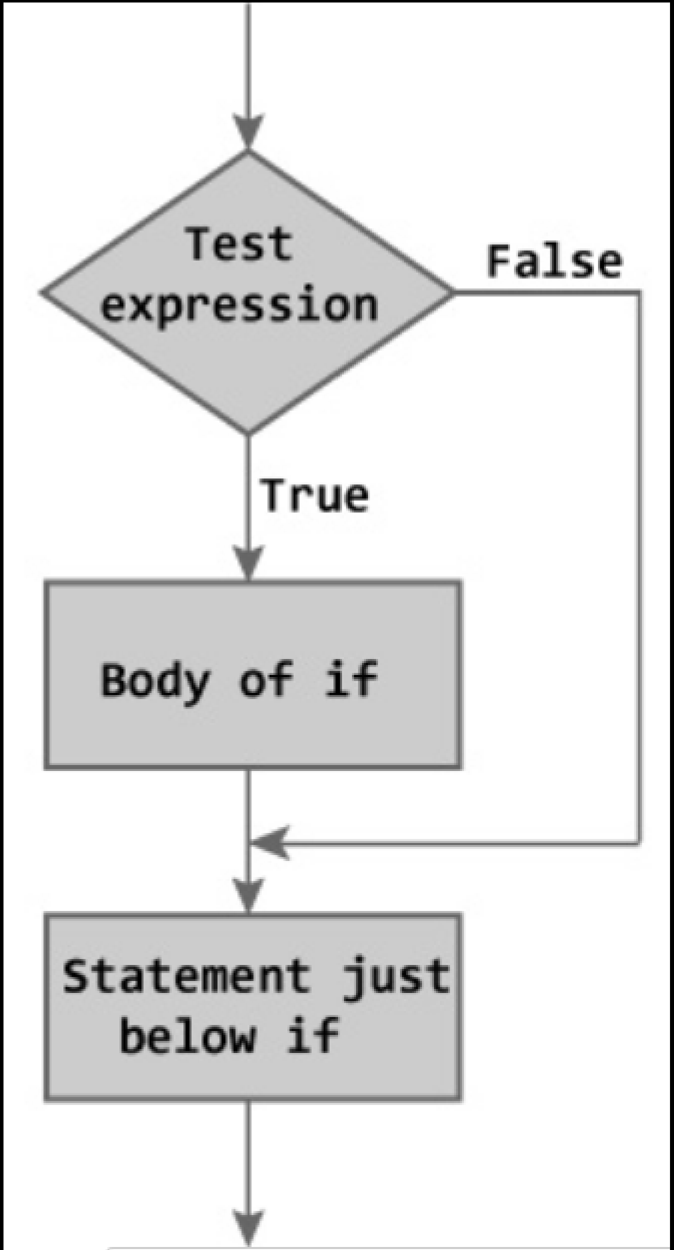
```
#include <stdio.h>
int main()
{
  int number;
  printf(“Enter a number ”);
  scanf(“%d”, &number);
  if(number < 0){
    printf(“Negative\n”);
  }
  else if(number > 0){
    printf(“Positive\n”);
  }
  else{
    printf(“Aha! It’s Zero!!\n);
  }
  printf(“Go do it\n”);
  return 0; }
```



Single-way Selection

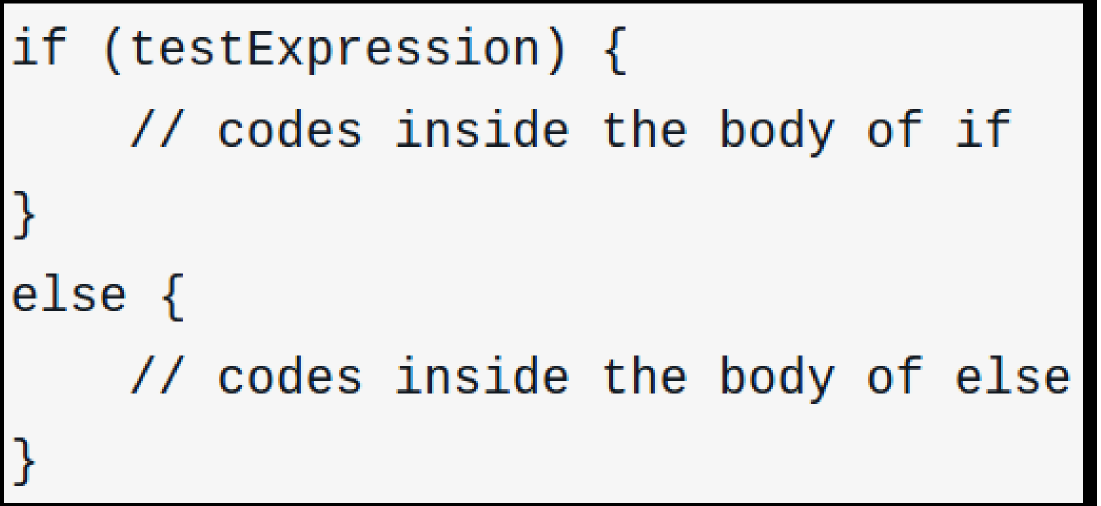


```
if (testExpression)
{
   // statements
}
```

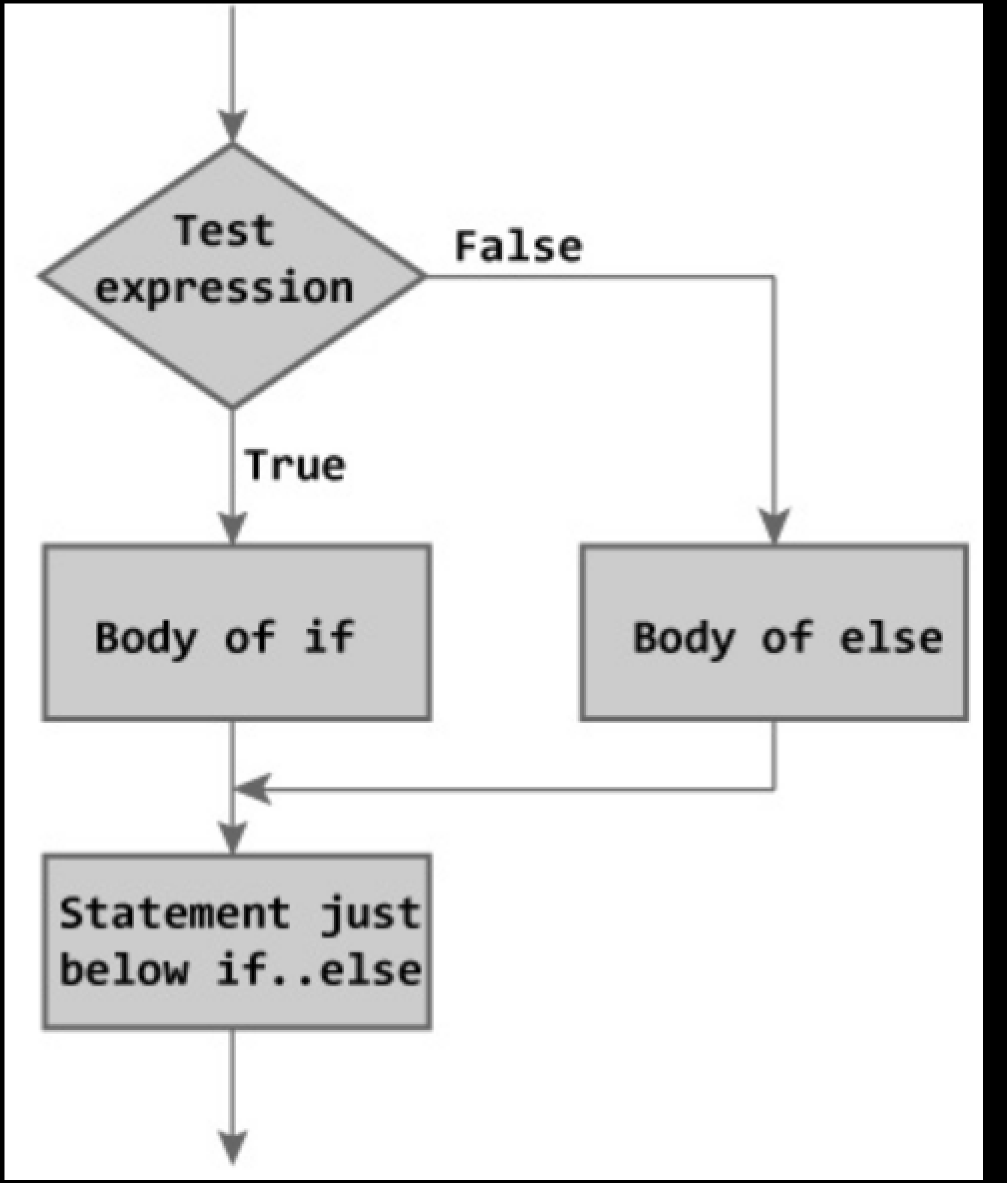


Credits: programiz.com

Two-way Selection

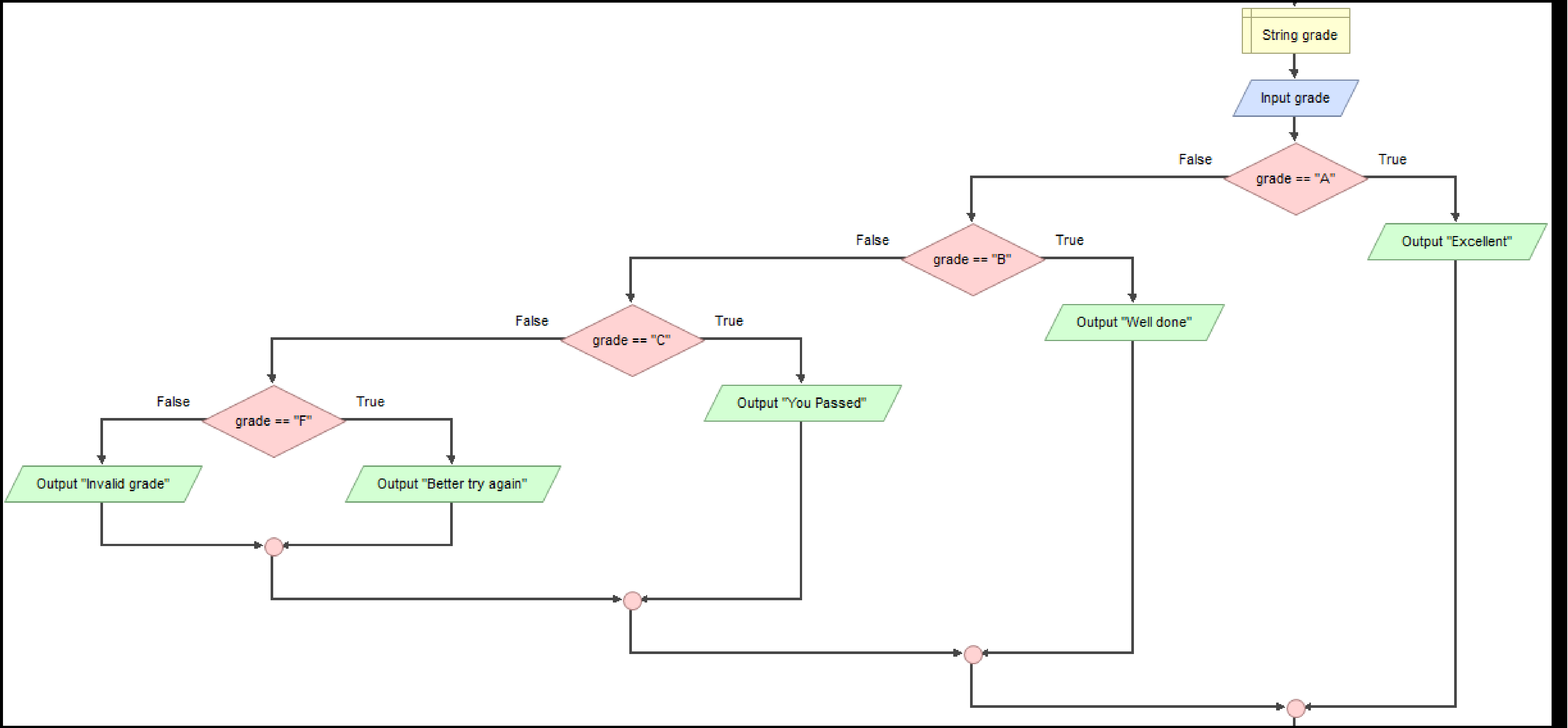


```
if (testExpression) {
    // codes inside the body of if
}
else {
    // codes inside the body of else
}
```

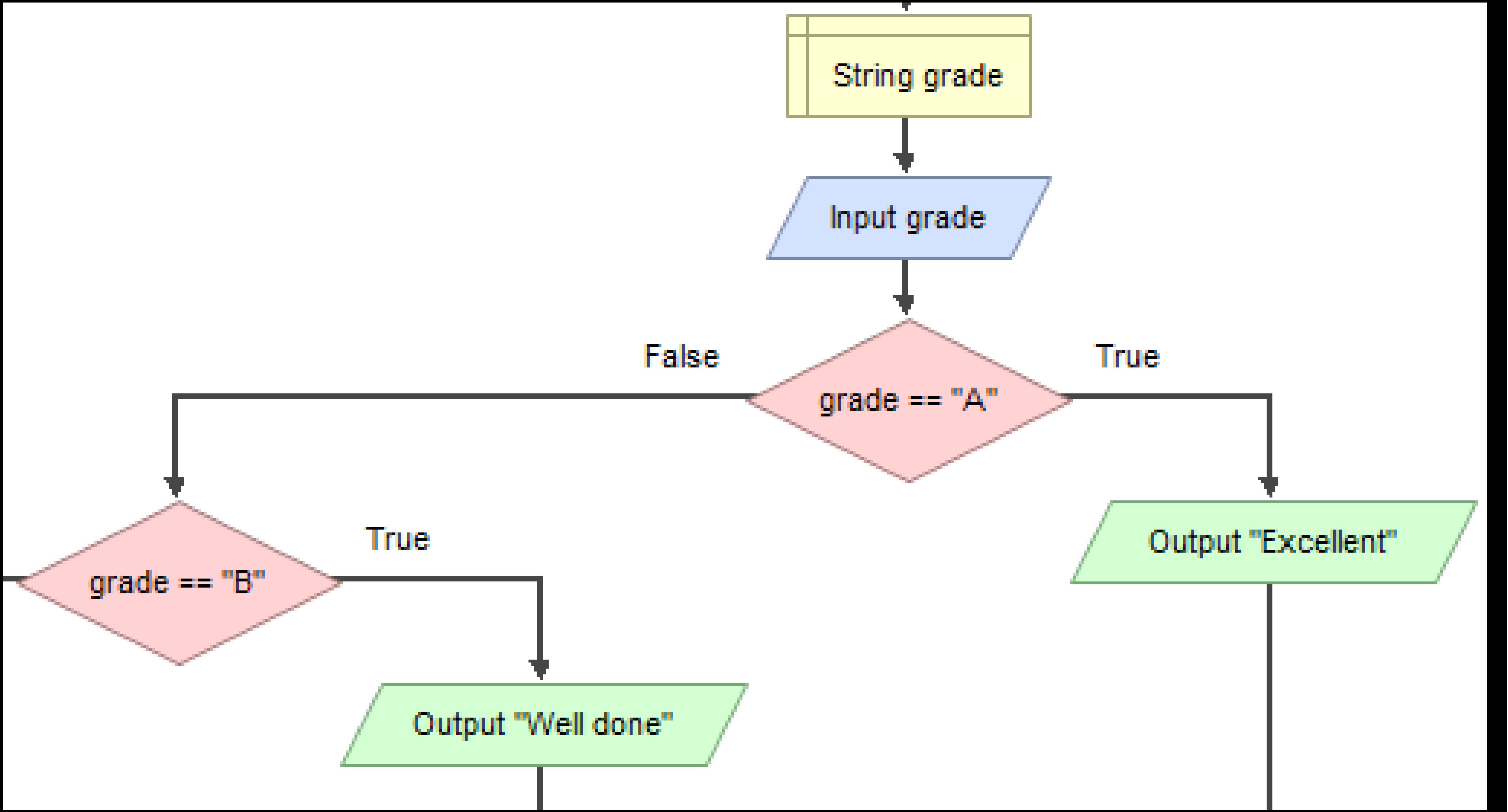


Credits: programiz.com

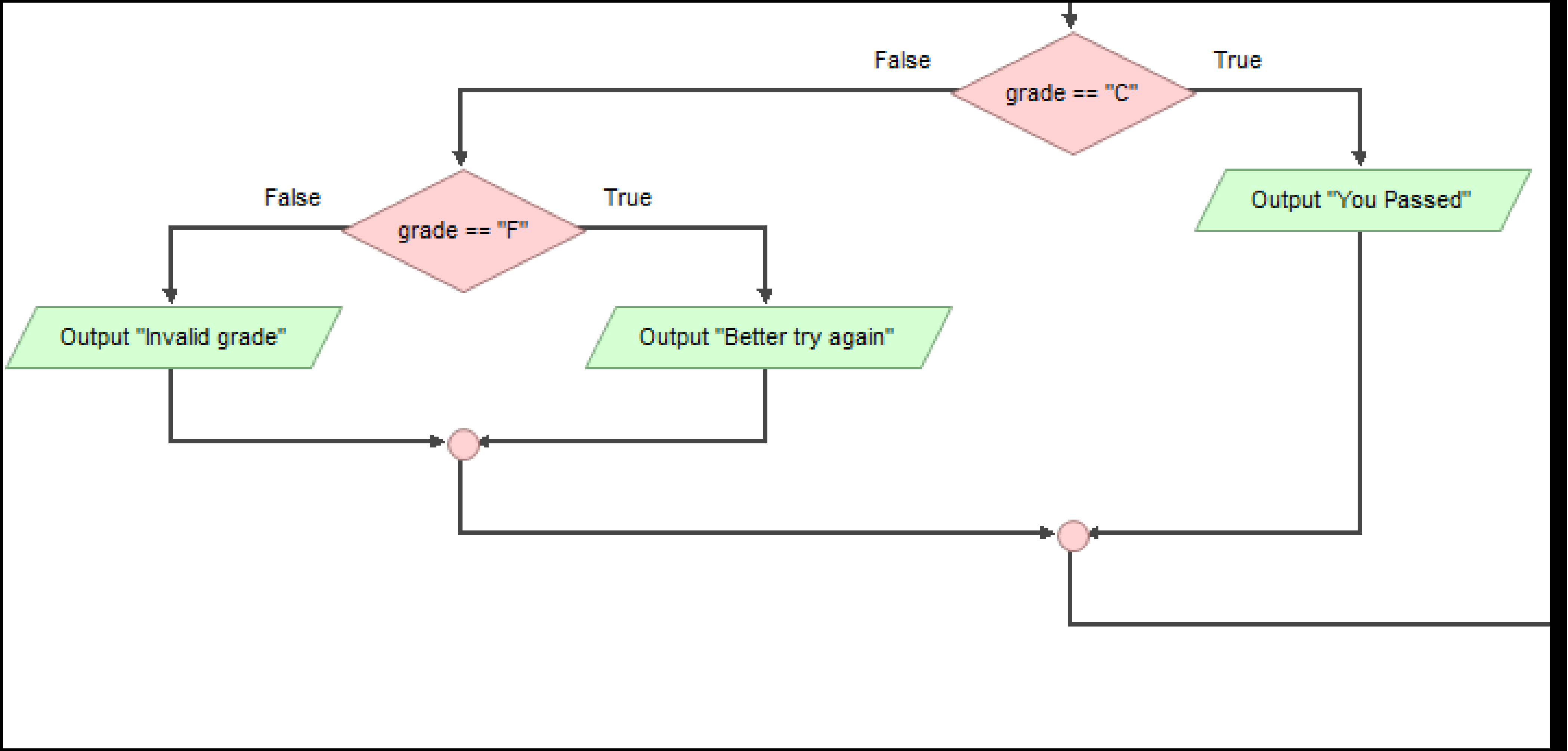
Multi-way Selection



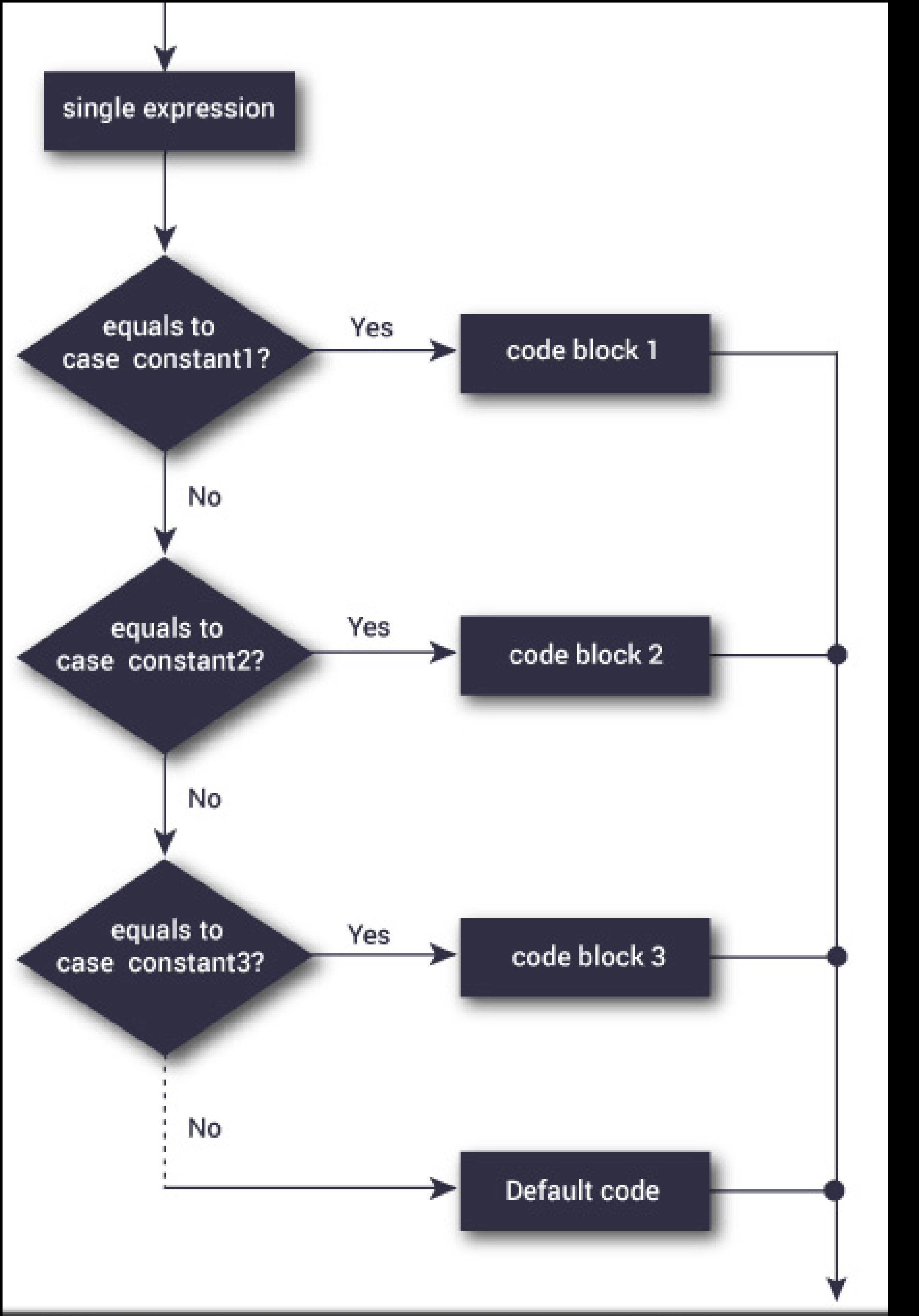
Multi-way Selection



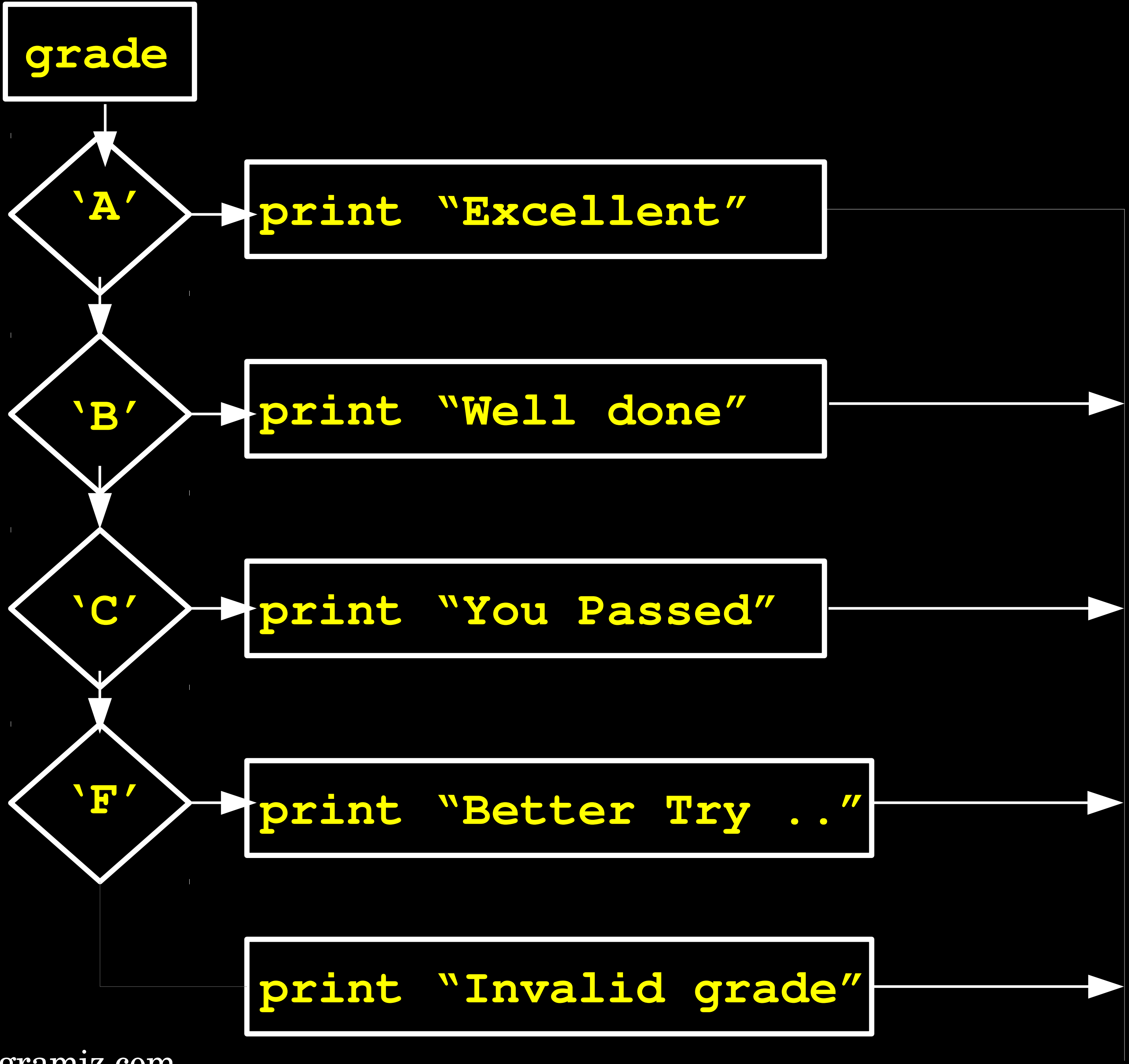
Multi-way Selection



Multi-way Selection



Credits: programiz.com



grade

‘A’

print “Excellent”

‘B’

print “Well done”

‘C’

print “You Passed”

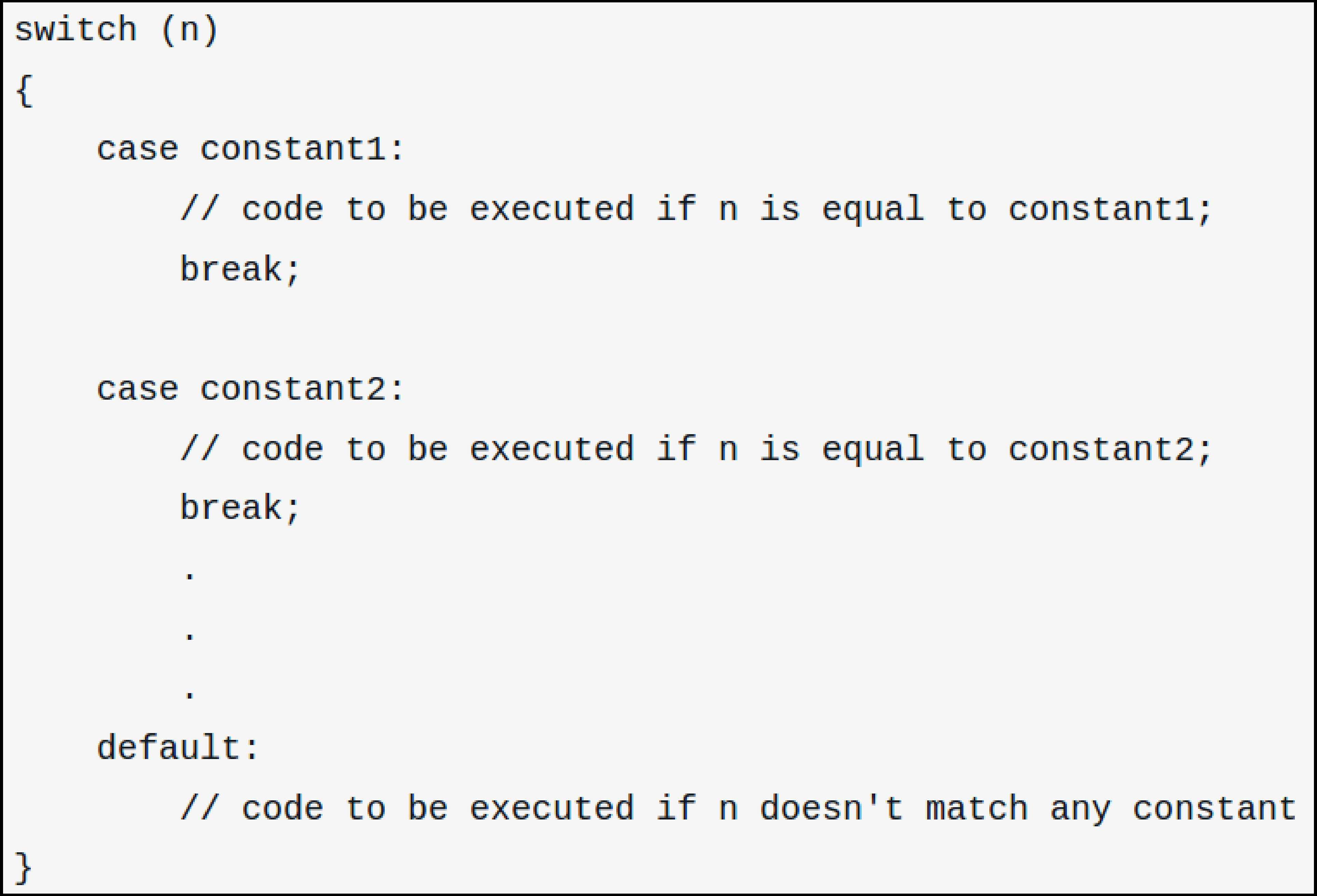
‘F’

print “Better Try ..”

print “Invalid grade”

Credits: programiz.com

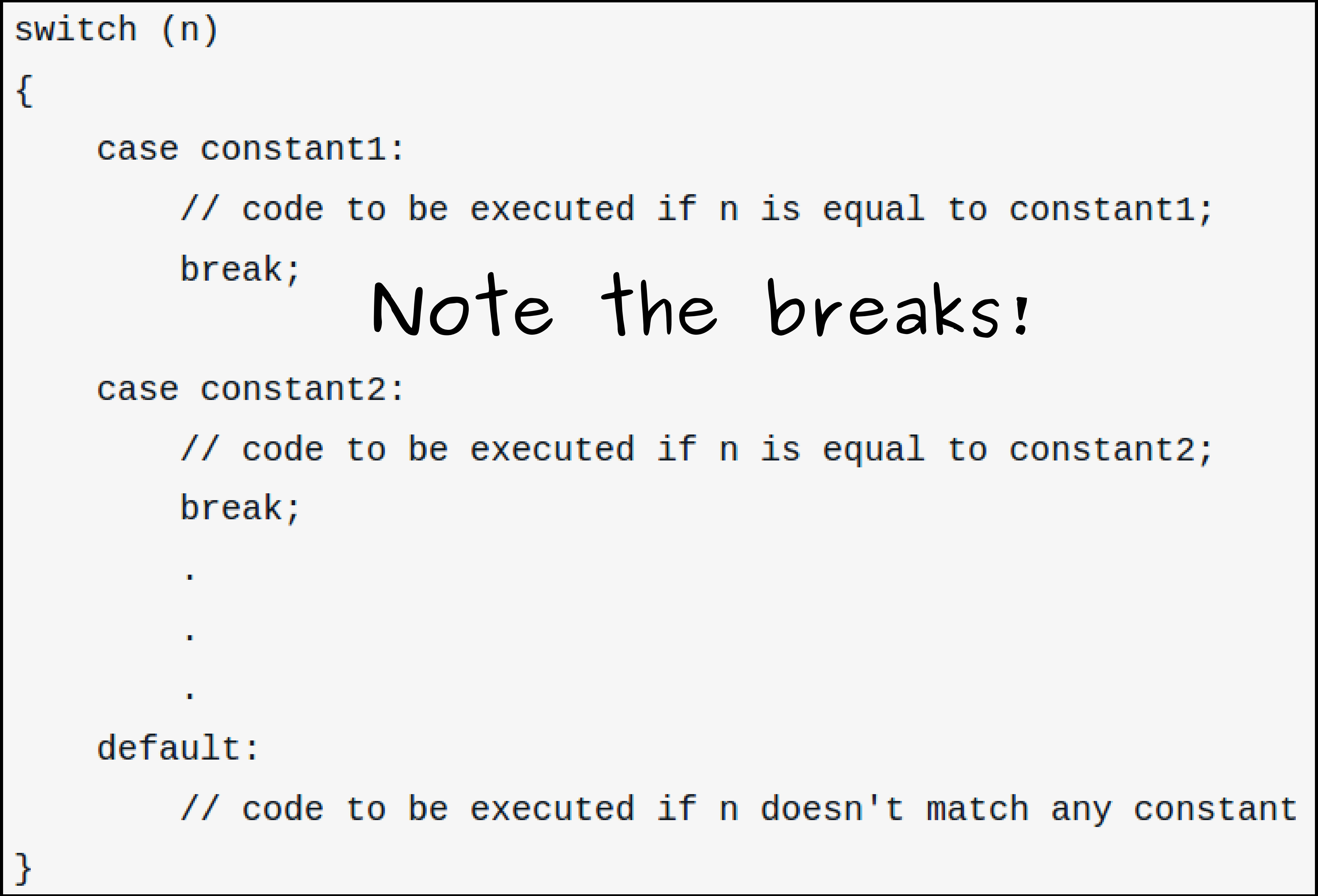
switch statement



```
switch (n)
{
    case constant1:
        // code to be executed if n is equal to constant1;
        break;

    case constant2:
        // code to be executed if n is equal to constant2;
        break;
        .
        .
        .
    default:
        // code to be executed if n doesn't match any constant
}
```

Credits: programiz.com

switch statement

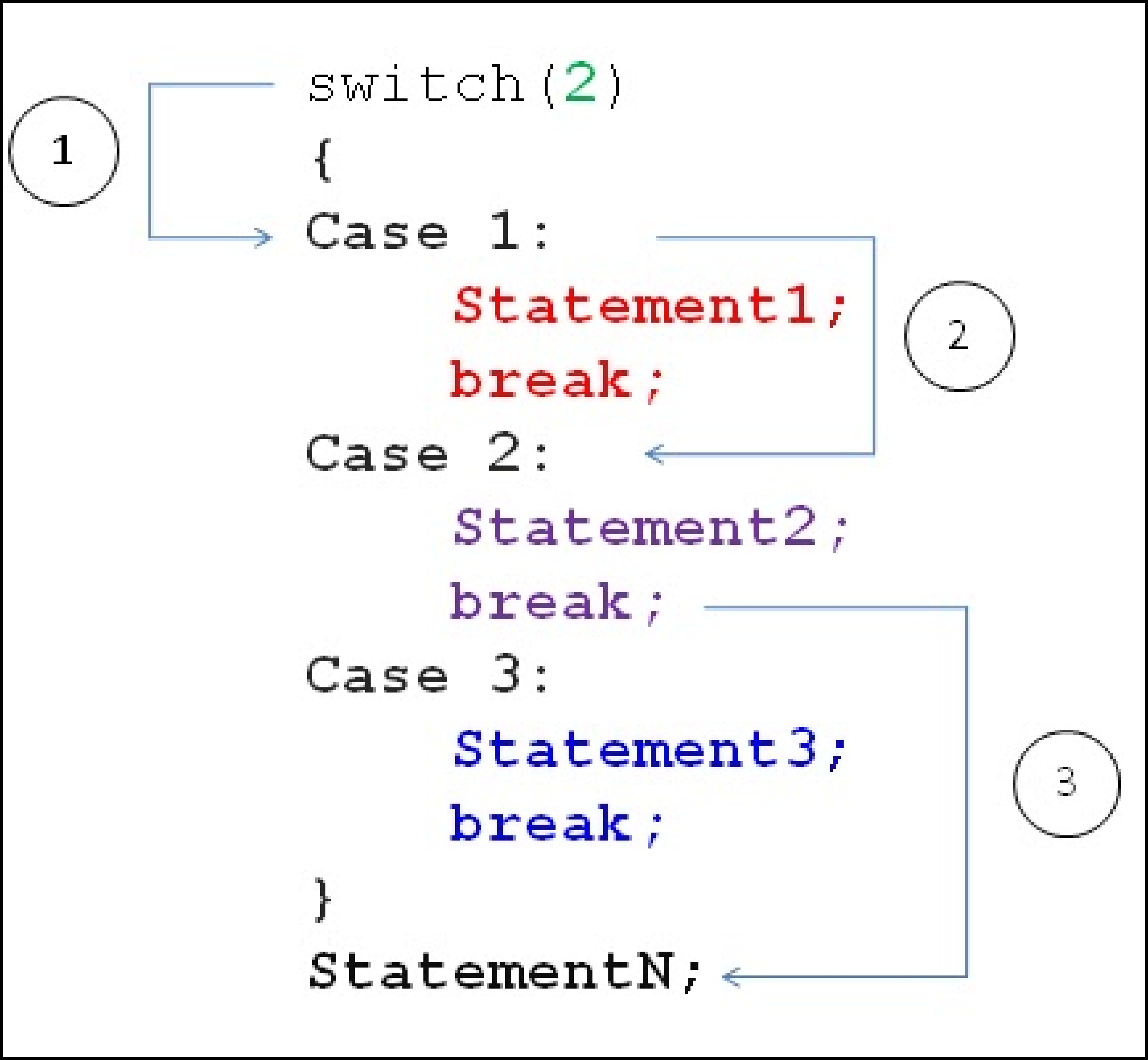


```
switch (n)
{
    case constant1:
        // code to be executed if n is equal to constant1;
        break;

    case constant2:
        // code to be executed if n is equal to constant2;
        break;
        .
        .
        .
    default:
        // code to be executed if n doesn't match any constant
}
```

Note the breaks!

Credits: programiz.com

switch Flow



```
switch(2)
{
Case 1:
        Statement1;
        break;
Case 2:
        Statement2;
        break;
Case 3:
        Statement3;
        break;
}
StatementN;
```

Credits: C4learn.com

```
switch(grade){
   case ‘A’:
      printf(“Excellent\n”);
      break;
   case ‘B’:
      printf(“Well done\n”);
      break;
   case ‘C’:
      printf(“You passed\n”);
      break;
   case ‘F’:
      printf(“Excellent\n”);
      break;
   default:
      printf(“Invalid grade\n”);
}
```

```
switch(grade){
  case ‘A’:
    printf(“Excellent\n”);
    break;
  case ‘B’:
    printf(“Well done\n”);
    break;
  case ‘C’:
    printf(“You passed\n”);
    break;
  case ‘F’:
    printf(“Excellent\n”);
    break;
  default:
    printf(“Invalid grade\n”);
}
```

switch is a keyword

```
switch(grade){
   case ‘A’:
      printf(“Excellent\n”);
      break;
   case ‘B’:
      printf(“Well done\n”);
      break;
   case ‘C’:
      printf(“You passed\n”);
      break;
   case ‘F’:
      printf(“Excellent\n”);
      break;
   default:
      printf(“Invalid grade\n”);
}
```

switch is a keyword

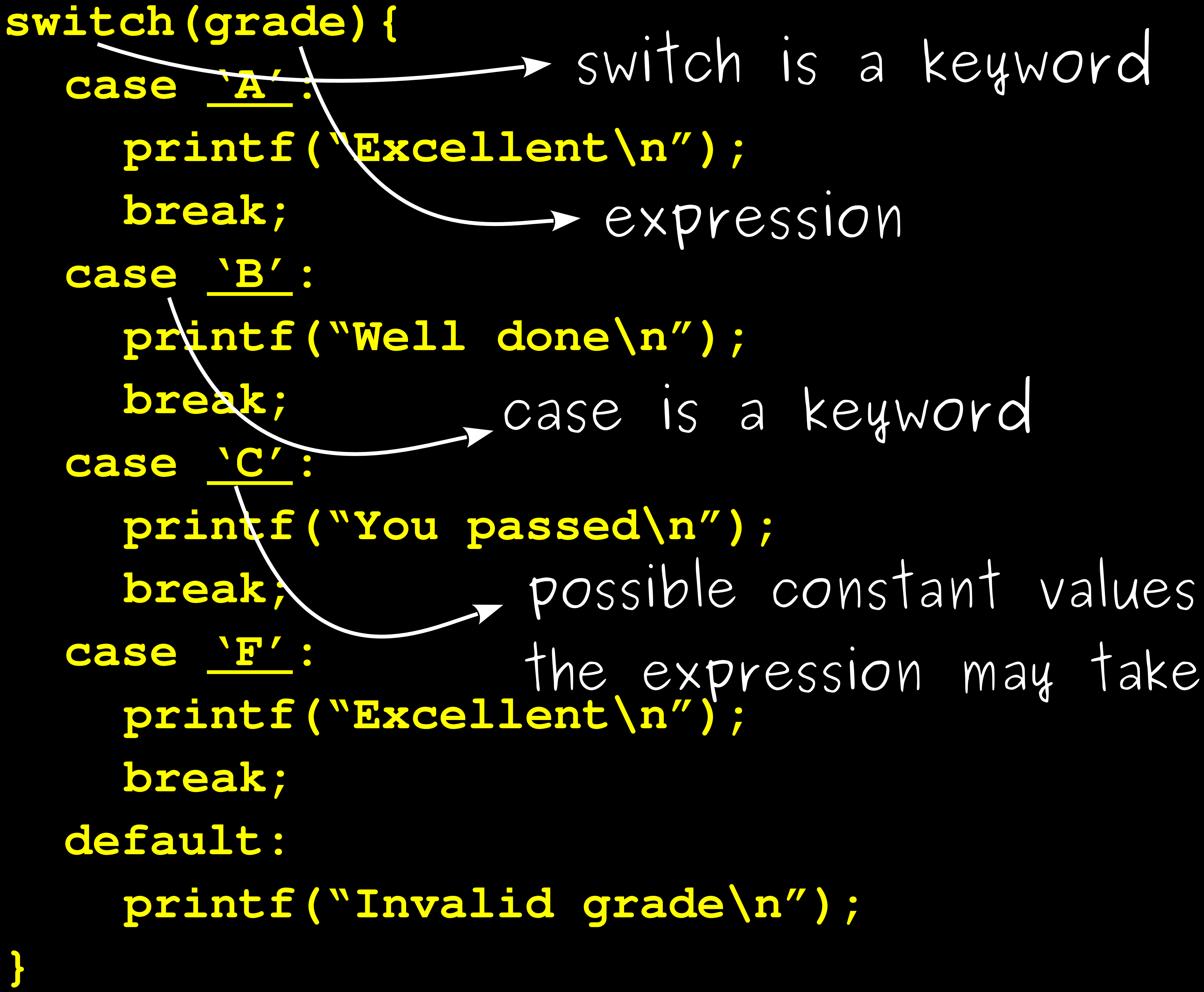
expression

```
switch(grade){
   case ‘A’:
      printf(“Excellent\n”);
      break;
   case ‘B’:
      printf(“Well done\n”);
      break;
   case ‘C’:
      printf(“You passed\n”);
      break;
   case ‘F’:
      printf(“Excellent\n”);
      break;
   default:
      printf(“Invalid grade\n”);
}
```

switch is a keyword

expression

case is a keyword; as many possible values of expression as many cases



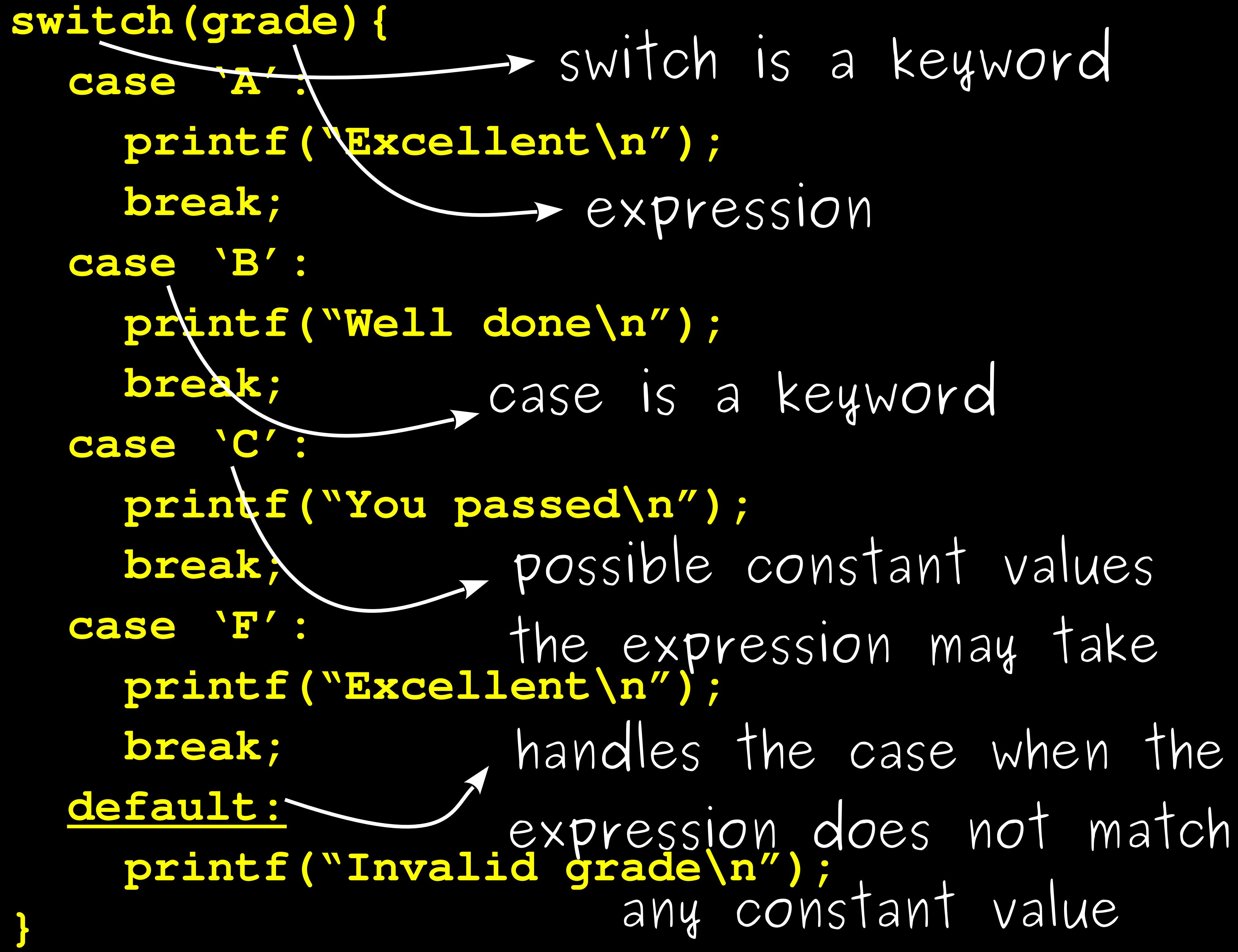
```
switch(grade){
   case ‘A’:
      printf(“Excellent\n”);
      break;
   case ‘B’:
      printf(“Well done\n”);
      break;
   case ‘C’:
      printf(“You passed\n”);
      break;
   case ‘F’:
      printf(“Excellent\n”);
      break;
   default:
      printf(“Invalid grade\n”);
}
```

switch is a keyword

expression

case is a keyword

possible constant values
the expression may take



```
switch(grade){
  case ‘A’:
    printf(“Excellent\n”);
    break;
  case ‘B’:
    printf(“Well done\n”);
    break;
  case ‘C’:
    printf(“You passed\n”);
    break;
  case ‘F’:
    printf(“Excellent\n”);
    break;
  default:
    printf(“Invalid grade\n”);
}
```

switch is a keyword

expression

case is a keyword

possible constant values the expression may take

handles the case when the expression does not match any constant value

```
switch(grade){
   case ‘A’:
      printf(“Excellent\n”);
      break;
   case ‘B’:
      printf(“Well done\n”);
      break;
   case ‘C’:
      printf(“You passed\n”);
      break;
   case ‘F’:
      printf(“Excellent\n”);
      break;
   default:
      printf(“Invalid grade\n”);
}
```

What is purpose of breaks?!

```
switch(grade){
  case ‘A’:
  case ‘a’:
    printf(“Excellent\n”);
    break;
  case ‘B’:
  case ‘b’:
    printf(“Well done\n”);
    break;
```

multiple cases are
like logical OR

```
  case ‘C’:
  case ‘c’:
    printf(“You passed\n”);
    break;
  case ‘F’:
  case ‘f’:
    printf(“Excellent\n”);
    break;
  default:
    printf(“Invalid grade\n”);
}
```

switch Template

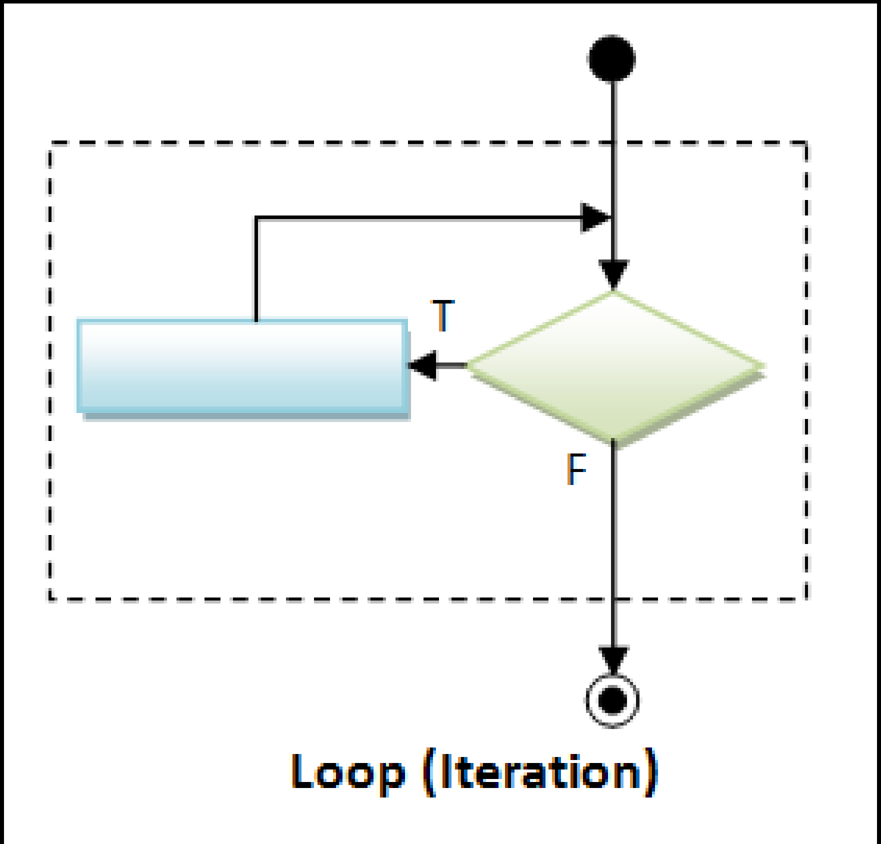
```
switch (□){
  case □:
    □
    break;
  .
  .
  default:
    □
}
```

switch Template

```
switch (expression){
  case const value:
    statement(s)
    break;
  .
  .
  default:
    statement(s)
}
```

CSE102
Computer Programming

(Next Topic)



Credits: https://www3.ntu.edu.sg/home/ehchua/programming/cpp/cp1_Basics.html