

جزوه درس برنامه نویسی

فواد اسماعیلی

۳۰-۱۱-۲۰۲۳

تا وقتی ندانیم چه چیزی می‌خواهیم، هرگز نمی‌توانیم، برنامه آن را بنویسیم!
بنابراین اول به صورت کاملاً واضح آن چیزی که می‌خواهیم را بهتر است روی کاغذ برای خود تجسم کنیم.

۱ شرط

آیا فلان هست؟

```
if(TRUE) {  
    #my code  
}  
if(FALSE) {  
    # my code does not run  
}  
if(x==2) {  
  
} else if(x==3)
```

مثال: اعداد بخش پذیر بر 5

```
x<- 1234  
x_1 <- x%%10  
if(x_1 ==0 || x_1==5){  
    print("is")  
} else{  
    print("not")  
}
```

```
## [1] "not"
```

```
n<- 50  
nc <- seq_len(n)  
v<- 3  
for(i in nc) {  
    if(i %%v==0) {  
        print(i)  
    }  
}
```

```
## [1] 3
## [1] 6
## [1] 9
## [1] 12
## [1] 15
## [1] 18
## [1] 21
## [1] 24
## [1] 27
## [1] 30
## [1] 33
## [1] 36
## [1] 39
## [1] 42
## [1] 45
## [1] 48
```

```
my_f <- function(n,v) {
  nc <- seq_len(n)
  for(i in nc) {
    if(i %%v==0) {
      print(i)
    }
  }
}
my_f(40,5)
```

```
## [1] 5
## [1] 10
## [1] 15
## [1] 20
## [1] 25
## [1] 30
## [1] 35
## [1] 40
```

```
x <- c("a","b","jafar","ali")
for(i in x) {
  print(i)
}
```

```
## [1] "a"
## [1] "b"
## [1] "jafar"
## [1] "ali"
```

```
x <- 5
if(x %%2 == 0 && x >2) {
  print(FALSE)
} else if(x %%3 == 0 && x>3) {
  print(FALSE)
} else if(x %%4 == 0 && x>4) {
  print(FALSE)
} else if(x %%5 == 0 && x>5) {
  print(FALSE)
} else{
```

```
print(TRUE)
}
```

```
## [1] TRUE
```

```
x <- 11
for(i in 2:x) {
  if(i == x) {
    print(TRUE)
  } else if(x %% i==0) {
    print(FALSE)
    break
  }
}
```

```
## [1] TRUE
```

```
aval <- function(x) {
  if(x<2) {
    stop("x must be larger or equal to 2")
  }
  for(i in 2:x) {
    if(i == x) {
      return(TRUE)
    } else if(x %% i==0) {
      return(FALSE)
    }
  }
}
aval(54)
```

```
## [1] FALSE
```

```
aval(55)
```

```
## [1] FALSE
```

```
aval(56)
```

```
## [1] FALSE
```

```
aval(57)
```

```
## [1] FALSE
```

```
aval(58)
```

```
## [1] FALSE
```

```
!FALSE
```

```
## [1] TRUE
```

من نمیدانم تا کجا ادامه دارد!

```
i <- 54
```

```
print(i)
```

```
## [1] 54
```

```
upto_aval <- function(n) {
  while(!aval(n)){
    n <- n+1
  }
  return(n)
}
upto_aval(1000)
```

```
## [1] 1009
```

```
n <- 30
for(i in 2:n) {
  if(aval(i)) {
    print(i)
  }
}
```

```
## [1] 2
## [1] 3
## [1] 5
## [1] 7
## [1] 11
## [1] 13
## [1] 17
## [1] 19
## [1] 23
## [1] 29
```

```
n <- 10
res_mat <- matrix(0,nrow =n,ncol = n)
for(i in 1:n) {
  for(j in 1:n) {
    res_mat[i,j] <- i*j
  }
}
res_mat
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    2    3    4    5    6    7    8    9    10
## [2,]    2    4    6    8   10   12   14   16   18   20
## [3,]    3    6    9   12   15   18   21   24   27   30
## [4,]    4    8   12   16   20   24   28   32   36   40
## [5,]    5   10   15   20   25   30   35   40   45   50
## [6,]    6   12   18   24   30   36   42   48   54   60
## [7,]    7   14   21   28   35   42   49   56   63   70
## [8,]    8   16   24   32   40   48   56   64   72   80
## [9,]    9   18   27   36   45   54   63   72   81   90
## [10,]   10   20   30   40   50   60   70   80   90   100
```

```
n <- 10
res_mat <- matrix(0,nrow =n,ncol = n)
for(i in 1:n) {
  res_mat[i,] <- i*seq_len(n)
}
res_mat
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    2    3    4    5    6    7    8    9   10
## [2,]    2    4    6    8   10   12   14   16   18   20
## [3,]    3    6    9   12   15   18   21   24   27   30
## [4,]    4    8   12   16   20   24   28   32   36   40
## [5,]    5   10   15   20   25   30   35   40   45   50
## [6,]    6   12   18   24   30   36   42   48   54   60
## [7,]    7   14   21   28   35   42   49   56   63   70
## [8,]    8   16   24   32   40   48   56   64   72   80
## [9,]    9   18   27   36   45   54   63   72   81   90
## [10,]   10   20   30   40   50   60   70   80   90  100
```

```
print("*")
```

```
## [1] "*"
```

```
print("**")
```

```
## [1] "**"
```

```
print("***")
```

```
## [1] "****"
```

```
print(rep("*",1))
```

```
## [1] "*"
```

```
print(rep("*",2))
```

```
## [1] "*" "*"
```

```
print(rep("*",3))
```

```
## [1] "*" "*" "*"
```

```
for(i in 1:5) {
  print(paste(rep("*",i),collapse = " "))
}
```

```
## [1] "*"
```

```
## [1] "*" "*"
```

```
## [1] "*" * "*"
```

```
## [1] "*" * * "*"
```

```
## [1] "*" * * * "*"
```

```
n<- 5
```

```
for(i in 1:n) {
  print(paste(rep(c("*", " "),c(i,n-i)),collapse = " "))
}
```

```
## [1] "*"      "
```

```
## [1] "*" *    "
```

```
## [1] "*" * *  "
```

```
## [1] "*" * * * "
```

```
## [1] "*" * * * * "
```

```
n<- 5
```

```
for(i in 0:n) {
  print(paste(rep(c(" ", "*", " "),c(n-i,1+2*i,n-i)),collapse = " "))
}
```

```
## [1] "          *          "
## [1] "          * * *      "
## [1] "          * * * * *   "
## [1] "          * * * * * * "
## [1] " * * * * * * * * * * "
## [1] "* * * * * * * * * *"
```

برنامه ای بنویسیم! که دو تا آرگومان داره از x شروع شود و تا y برود و به محضی که به عدد اول رسید چاپ کند و خارج شود.

```
x <- 33
y <- 50
for(i in x:y) {
  if(aval(i)){
    print(i)
    break
  }
}
```

```
## [1] 37
```

```
my_f <- function(x,y) {
  for(i in x:y) {
    if(aval(i)){
      return(i)
    }
  }
  print("no value")
}
my_f(100004,100019)
```

```
## [1] 100019
```

$\text{mean}(1:i)/\text{sd}(1:i)$

```
mean(1:2)/sd(1:2)
```

```
## [1] 2.12132
```

```
mean(1:3)/sd(1:3)
```

```
## [1] 2
```

```
mean(1:4)/sd(1:4)
```

```
## [1] 1.936492
```

```
mean(1:5)/sd(1:5)
```

```
## [1] 1.897367
```

```
mean(1:6)/sd(1:6)
```

```
## [1] 1.870829
```

```
n <- 6
for(i in 2:n) {
  print(mean(1:i)/sd(1:i))
}
```

```
## [1] 2.12132
```

```
## [1] 2
## [1] 1.936492
## [1] 1.897367
## [1] 1.870829
```

```
n <- 20
x <- rnorm(n)
```

$$\sum_{i=2}^n X_i$$

$$X_i = \text{mean}(x[1:i]) / \text{sd}(x[1:i])$$

```
res <- c()
for(i in 2:n) {
  res[i-1] <- mean(x[1:i]) / sd(x[1:i])
}
sum(res)
```

```
## [1] 2.004834
```