CENG 230

Introduction to C Programming

Week 7 – Repetition

Sinan Kalkan

Some slides/content are borrowed from Tansel Dokeroglu, Nihan Kesim Cicekli.
Conditional Expression Operator

\[ y = x > 3 \ ? \ a+1 \ : \ a-1; \] means

\[
\begin{align*}
\text{If } (x > 3) \\
y &= a + 1; \\
\text{else} \\
y &= a - 1;
\end{align*}
\]

\[ z = (a > b) \ ? \ a : b; \quad \text{(finds maximum)} \]

\[
\text{Printf(“%d%c”, k, (k\%10==9) ? ’A’ : ’a’);}
\]
Loops, iterations, repetitions
while, do-while and for statements
Most programs involve repetition or looping.

A **loop** is a group of instructions the computer executes repeatedly while some **loop-continuation condition** remains true.

```c
/* Fig. 4.1: fig04_01.c
Counter-controlled repetition */
#include <stdio.h>

/* function main begins program execution */
int main( void )
{
    int counter = 1; /* initialization */

    while ( counter <= 10 ) { /* repetition condition */
        printf( "%d\n", counter ); /* display counter */
        ++counter; /* increment */
    } /* end while */
    return 0; /* indicate program ended successfully */
} /* end function main */
```
- While statement
- Do...While statement
- For statement
Repetitions

• while loop

  Initialization;
  while( expr )
   statement;

  Initialization;
  while( expr )
  {
    statement;
    statement;
    statement;
  }

• Bad examples:
  while( x = 1 )
  {
    x = getchar();
  }

  x = 0.0;
  while( x != 1.0 )
  {
    x += 0.005;
  }
Example

• Factorial

```c
int N, fact = 1;
scanf("%d", &N);
while( N > 0 )
{
    fact *= N--;
}
```
Repetitions

do-while loop

Initialization;
do
statement
while( expr );
statement;

Initialization;
do
{
    statement;
    statement;
    statement;
} while( expr );


do
{
    x = getchar();
    putchar(x);
} while( x != EOF );
Example

• Factorial with do-while:

```c
int N, fact = 1;
scanf("%d", &N);
do
  { fact *= N--; }
while( N > 0 );
```
Finding power of a number

/* C program to calculate the power of an integer*/
#include <stdio.h>
int main()
{
    int base, exp;
    long long int value=1;
    printf("Enter base number and exponent respectively: ");
    scanf("%d%d", &base, &exp);
    while (exp!=0)
    {
        value*=base; /* value = value*base; */
        --exp;
    }
    printf("Answer = %d\n", value);
    system("pause");
}
Today

• Continue with repetitions
  • More examples
  • “for” loops
#include <stdio.h>
int main()
{
    int count, n, t1=0, t2=1, display=0;
    printf("Enter number of terms: ");
    scanf("%d", &n);
    printf("Fibonacci Series: %d\n%d\n", t1, t2); /* Displaying first two terms */
    count=2;    /* count=2 because first two terms are already displayed. */
    while (count<n)
    {
        display=t1+t2;
        t1=t2;
        t2=display;
        ++count;
        printf("%d \n", display);
    }
    printf("%d \n", display);
    system("pause");
    return 0;
}
Repetitions

• **for** loop

  *Initialization;*

  for( expr1; expr2; expr3 )
  statement

  *Initialization;*

  for( expr1; expr2; expr3 )
  {
    statement;
    statement;
    statement;
  }

  for( j = 0; j < N; j++)
  printf("j: %d\n", j);

  for( i=0, j=0;
      i < 0 & j > N; i++, j--);

  for(   ;   ; i++ )
  {
    if( i > 0 ) return 0;
  }
/* function main begins program execution */
int main( void )
{
    int counter; /* define counter */

    /* initialization, repetition condition, and increment are all included in the for statement header. */
    for ( counter = 1; counter <= 10; counter++ ) {
        printf( "%d\n", counter );
    } /* end for */

    return 0; /* indicate program ended successfully */
} /* end function main */
The diagram illustrates the structure of a `for` loop in programming. The `for` keyword is followed by the control variable name, a semicolon, the initial value of the control variable, another semicolon, the condition in which the loop will continue, another semicolon, and the increment of the control variable. The example given is:

```plaintext
for (counter = 1; counter <= 10; counter++)
```
1. Vary the control variable from 1 to 100 in increments of 1.
   \[ \textbf{for} \ ( \ i = 1; \ i <= 100; \ i++ ) \]

2. Vary the control variable from 100 to 1 in increments of -1 (decrements of 1).
   \[ \textbf{for} \ ( \ i = 100; \ i >= 1; \ i-- ) \]

3. Vary the control variable from 7 to 77 in steps of 7.
   \[ \textbf{for} \ ( \ i = 7; \ i <= 77; \ i += 7 ) \]

4. Vary the control variable from 20 to 2 in steps of -2.
   \[ \textbf{for} \ ( \ i = 20; \ i >= 2; \ i -= 2 ) \]

5. Vary the control variable over the following sequence of values: 2, 5, 8, 11, 14, 17.
   \[ \textbf{for} \ ( \ j = 2; \ j <= 17; \ j += 3 ) \]

6. Vary the control variable over the following sequence of values: 44, 33, 22, 11, 0.
   \[ \textbf{for} \ ( \ j = 44; \ j >= 0; \ j -= 11 ) \]
/* Fig. 4.5: fig04_05.c
Summation with for */
#include <stdio.h>

/* function main begins program execution */
int main( void )
{
    int sum = 0; /* initialize sum */
    int number; /* number to be added to sum */

    for ( number = 2; number <= 100; number += 2 ) {
        sum += number; /* add number to sum */
    } /* end for */

    printf( "Sum is %d\n", sum ); /* output sum */
    return 0; /* indicate program ended successfully */
} /* end function main */

Sum is 2550
Nested Loops

• You can have loops within loops:

```plaintext
for(i=0; i<N; i++)
{
    for(j=0; j<N; j++)
    {
        ...
    }
}
```
```c
#include <stdio.h>

/* function main begins program execution */
int main( void )
{
    int x;
    int y;
    int i;
    int j;

    /* prompt user for input */
    printf( "Enter two integers in the range 1-20: " );
    scanf( "%d%d", &x, &y ); /* read values for x and y */

    for ( i = 1; i <= y; i++ ) { /* count from 1 to y */
        for ( j = 1; j <= x; j++ ) { /* count from 1 to x */
            printf( "@" ); /* output @ */
        } /* end inner for */
    } /* end outer for */

    printf( "\n" ); /* begin new line */
} /* end function main */
```
4.36 What does the following program segment do?

```
for ( i = 1; i <= 5; i++ ) {
    for ( j = 1; j <= 3; j++ ) {
        for ( k = 1; k <= 4; k++ )
            printf( "*" );
        printf( "\n" );
    }
    printf( "\n" );
}
```
infinite loops
(loops that do not finish executing)

```c
#include <stdio.h>

/* function main begins program execution */
int main( void )
{
    int counter = 1; /* initialization */

    while ( 1 ) { /* repetition condition */
        printf ( "%d\n", counter ); /* display counter */
        counter++; /* increment */
    } /* end while */

    system("pause");
    return 0; /* indicate program ended successfully */
} /* end function main */
```
Factors of a number

```c
#include <stdio.h>
int main()
{
    int n, i;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    printf("Factors of %d are: ", n);
    for (i = 1; i <= n; ++i)
    {
        if (n % i == 0)
            printf("%d ", i);
    }
    system("pause");
    return 0;
}
```
break;

- Stop the loop/iteration and continue with the statement after the loop.
- Usable with while, for and do-while

```c
while(...) {
    ... break;
    ...
}
```

```c
while( 1 ) {
    c = getchar();
    if( c == EOF )
        break;
    putchar( c );
}
```
continue;

- Skips the remaining statements in the loop and continues with the “loop head”.
- Usable with while, for and do-while

```c
Sum = 0;
for(i=0; i<N; i++)
{
    if( i%2 == 0 )
        continue;
    sum = sum + i;
}
```
Homework

• Write a program to read in numbers until the number -1 is encountered. The sum, max and min of all numbers read until this point should be printed out.
41) What is the output?
   
   ```c
   for(i=0; i<=2; i++)
   for(j=1; j<3; j++)
       printf("%d%d", i, j);
   printf("%d%d", i, j);
   ```
   a) 01021112212233  
   b) 0102111221222  
   c) 011121021222  
   d) 01112102122233  
   e) 01112102122222

42) What is the output?

   ```c
   int n=0, i=9, j=0;
   for(i=1, j=7; i<=j; i++, j--)
       n++;
   printf("%d%d%d", i, j, n);
   ```
   a) 170  
   b) 443  
   c) 444  
   d) 534  
   e) 900
43) What is the output?

```c
int k=456;
float t=0;
while(k/100>4){
    t=t+k/100;
    k=k-100;
}
printf("%f",t);
```

a) 0.000000   b) 4.000000   c) 4.560000

d) 56.000000   e) infinite loop

44) What is the output?

```c
for(i=0; i<9 ; i++){
    printf("%d",i);
    for(j=0; j<2; j++)
        i=i+2;
}
```

a) 012345678   b) 036   c) 048

d) 05   e) compile-time error

45) What is the output?

```c
for(i=2;i<10;i++) {
    if (i%3==0) continue;
    if (i%6==0) break;
    printf("%d",i);
}
```

a) 2345   b) 245   c) 24578   d) 3   e) 39
46) for(i=0; i<4; i++){
    for(j=0; j<4-i; j++)
        printf("%d ", ___(1)___);
    printf("\n");
}

Which expression should be replaced with ___(1)___ for this output:

0 3 6 9
2 5 8
4 7
6

a) i*(i+1)+3*j  b) 3*i+2*j  c) 2*i+3*j
d) (i+j)*3  e) 3*i+j*(j+1)

47) What will be the output when the input below?

Input: 200 1000 4 30 -1

int n, min=50000, tot=0;
do {
    scanf("%d", &n);
    if (n<min) min=n;
    tot=tot+min;
} while (n!=-1);
printf("%d %d", min, tot);

a) 4 204  b) 4 408  c) -1 203  d) 4 1234  e) -1 407
48) What is the output?
    int b=1;
    while(b<10 && b>-10) {
        b=b*2;
        printf("%d ",b);
    }

a) 1 -2 4 -8  
  b) -2 4 -8 16  
  c) 1 -2 4 -8 16  
  d) -2 4 -8  
  e) no output

49) How many DONE will be printed with the input 5 ?

    int i;
    scanf("%d",&i);
    do{
        printf("DONE");
    } while(i<10);

a) 0  
  b) 1  
  c) 5  
  d) infinite  
  e) 10

50) What is the output?
    int i=0, j=0;
    do{
        for(i =0 ; i< 5 ; i++)
            j+=i;
    }while(j<10);
    printf("%d %d", i, j);

a) 10 10  
  b) 0 10  
  c) 10 5  
  d) 5 10  
  e) infinite loop
39) Which of the following displays “hello” 5 times?

a) for( i=-1; i<=2; i+=1) printf(" hello ");

b) for(i=1; i<6; i+=2) printf(" hello ");

c) for( i=12; i<=16; i+=1) printf(" hello ");

d) for( i=0; i<=4; i-=1) printf(" hello ");

e) for( i=5; i<=0; i-=1) printf(" hello ");

40) What is the output of the following code segment?

```c
s=5;
while(s<10)
{
    s += 2;
    printf("%3d", 2*s);
    s++;
}
```

a) 7 10  b) 14 20  c) 5 6  d) 10 7  e) None of them
41) If the following statements display “computers” 3 times, what should be the statement ____{(1)}____?

```c
int i = 7/2;
while (i <= 10)
{
    ++i;
    printf("computers");
    ____{(1)}____;
}
```

a) i=i+1;   b) i=i+2;   c) i=i+3;   d) i=i+4;   e) i=i+5

42) What is the output of the following code segment?
```c
for( i = 10; i > 4; i--)
{
    printf("%d ", i-2);
    i -= 3;
}
```

a) 7 5   b) 8 4   c) 10 8   d) 8 2   e) 8 10

43) Which loop outputs 0 1 2?

a) for(i=1/2; i<6; i+=2) printf("%3d", i-2);
b) for(i=2; i<6; i+=2) printf("%3d", i-3);
c) for(i=5/2-1; i<9/2; i+=1) printf("%3d", i-1);
d) for(i=0; i<6; i+=2) printf("%3d", i);
e) for(i=2; i<6; i+=1) printf("%3d", i+3);
44) What is the output of the following program segment?

```c
k=5;
m=10;
while( k > 0 )
{
    if( m%3 )
        printf("%3d", m--);
    else
        printf("%3d", --m);
    k *= 2;
}
printf("%3d", k);
```

a) 10 7 7 1
b) 9 9 7 -1
c) 10 8 8 -1
d) 9 8 7 -1
e) 9 9 8 -1

45) What is the output of the following code segment?

```c
int i, k;
k=5/2;
for (i=3;i<=10; i+=2)
{
    ++i;
    if(k=3&&i%2)
    
        printf("BBB");
    else
        printf("AAA");
    k++;
}
```

a) BBAAAA
b) AAABBBBB
   c) AAABB
   d) AAABBBAAA
   e) BBAAABBBB
46) How many times the condition is checked?
   i=1;
   k=5;
   while (i <= 10-k)
   {
     ++i;
     printf("%3d",i);
     k+=2;
   }
   
a) 1   b) 2   c) 3   d) 4   e) 5

47) What will the following program print?

   #include<stdio.h>
   int x,y;
   main()
   {
     for (x=1,y=1; x<5 && y<3; x=x+1, y=y+1) printf("*");
   }

   a) Nothing
   b) **
   c) ***
   d) *****
   e) *********(8 asterisk)
48) What will the following program print?

```c
#include <stdio.h>
int x, y;
main()
{
for (x=1; x<5; y=y+1)
for (y=x+1; y<5; x=x+1) printf("*");
}
a) the printing of * will not stop
b) **
c) ***
d) *****
e) ******** (8 asterisk)
```
49) What will the following program print?

```c
#include<stdio.h>
int x,a,b,c;
main()
{
for (a=5; a>=1; a=a-1)
for (b=1; b<=a; b=b+1)
for (c=1; c<=b; c=c+1) x = x+1;
printf("%d",x);
}
a) 18        b) 17        c) 35        d) 70        e) 140
```

50) Which is true for the given program?

```c
#include<stdio.h>
int i = 0, j = 0;
main() {
do { printf("%d ",i+j);
   if((i+j)%2) printf("%d ",i+j);
   i++; j++;
} while (i<=j<3);
}
a) Will go into an infinite loop.
b) Will output 0 2 4
c) Will output 0 2
d) Will produce a compile time error.
```
Use below program to answer questions 21-22.

```c
#include <stdio.h>
int main() {
    int a=0,b=0,c=0,f,g,h;
    scanf("%d%d%d", &f, &g, &h);
    for (a=g; a<f; a++)
        switch(a) {
            case 1: c++; break;
            default: c += 2;
        }
    printf("%d\n", c);
}
```

21- What is the output of the above program for the input 4 1 1?

a) 1  
   b) 2  
   c) 3  
   d) 4  
   e) 5

22- What is the output of the above program for the input 5 2 1?

a) 1  
   b) 4  
   c) 6  
   d) 10 
   e) 7

23- What is the output of the above program for the input 5 1 1?

a) 1  
   b) 4  
   c) 5  
   d) 7  
   e) 15
```c
#include <stdio.h>
int main() {
    int a=0, b=0, c=0, f, g;
    scanf("%d%d",&f, &g);
    c=0;
    for (a=g;a<f;a++)
        for (b=g;b<a;b++)
            c++;
    printf("%d\n",c);
}
```

24- one of the below is the output of the above program for the input 5 2?

a) 3  b) 6  c) 10  d) 12  e) 15

25- one of the below is the output of the above program for the input 6 3?

a) 3  b) 6  c) 10  d) 12  e) 15
32. What will be the output of the below code segment?
   
   ```c
   m=0;
   do {
       m=m-2;
   } while (m>5)
   printf("%d",m);
   ```

   a) 0  b) 2  c) -2  d) 5  e) 7

33. What will be the output of the below code segment?
   
   ```c
   m=0;
   while (m>5)
       m=m-2;
   printf("%d",m);
   ```

   a) 0  b) 2  c) -2  d) 5  e) 7

34. What will be the value of `dif` at the end of following code segment?
   
   ```c
   int m=1;
   int myvar,dif;
   while(m<=2)
       myvar=m++;
   dif=m-myvar;
   ```

   a) 0  b) 1  c) -1  d) 2  e) -2
Use below program to answer to questions 34-35.

```c
counter1 =0
counter2=0;
while (counter1 <3 ) {
    while ( (counter2+counter1)%2==0)
        printf("%d",counter2++);
    counter1++;
}
```

35. How many times will the printf statement be executed?

   a) 3    b) 4    c) 7    d) 0    e) 2

36. What will be the value of the `counter2` after the execution of the above code segment?

   a) 3    b) 0    c) 2    d) 4    e) 1
13. What will be the value of $x$ after the following program segment is executed?

```c
int i, x; x = 0; i = 100;
while (i > 0) {
    x++;
    i = i / 2;
}
```

a) 6  
b) 7  
c) 8  
d) 9

15. What will be the output of the following program segment?

```c
int n = 12, j = 2;
while (j <= n) {
    if (n % j == 0){
        n = n / j;
        printf("%d ", j);
    }
    else  j++;
}
```