C Language

Iterative Control instruction

Part - 2

Saurabh Shukla (MySirG)

Agenda

1. do-while loop
2. for loop
3. break
4. continue

Entry Control loop

```
while (condition)
{
   ==
}
```

```
int i=1;
while(i<=10)
{  printf("%d ",i);
   i++;
}
```

1<=10 T
2<=10 T
3<=10 T
...
10<=10 T
11<=10 F] 10

Exit Control loop

```
do
{
   ==
}while(condition);
```

```
int i=1;
do
{  printf("%d ",i);
   i++;
}while(i<=10);
```

2<=10 T
3<=10 T
4<=10 T
5<=10 T
...
10<=10 T
11<=10 F] 9

1
2
3
4
...
10

Entry Control loop

```
for( ; ; )
{
   ==
}
```

```
int i;
for(i=1; i<=10; i++)
{  printf("%d", i);
}
```

1<=10 T
2<=10 T
3<=10 T
...
10<=10 T
11<=10 F] 10

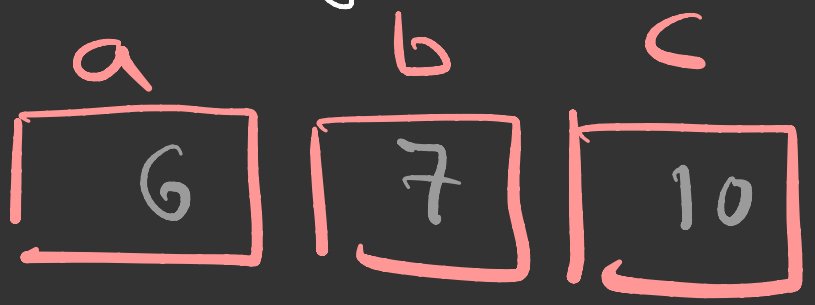
1
2
3
...
10

```
int main()
{
    int i=1;

    for(    ;    ;    )
    {
        printf("%d ", i);
        i++;
    }

}
```

while( ) error

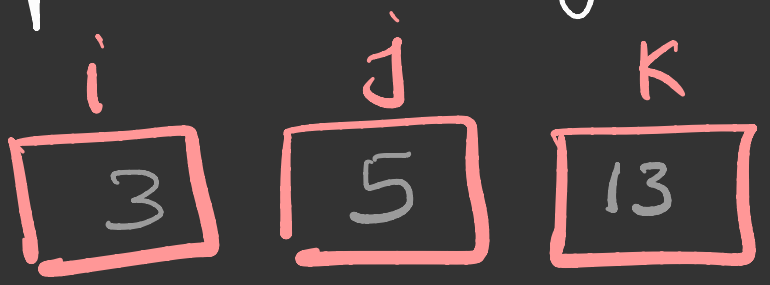
Find Output of the program

```
int main()
{
    int a=2, b=-1, c;
    do
    {
        c = a+b+1;
        printf("%d", c);
        a = b+1;
        b = a+1;
    } while(c<10);
}
```



2 2 6 10

Find output of the program




```
int main()
{
    int i, j, k;

    for (i=0; i<3; i++)
    {
        j = i*2+1;
        k= j*3-2;
    }
    printf("%d %d %d", i, j, k);

}
```

break

```
while(condition)
{
    ==
    ==
    break;
    ==
}
```

- break is a keyword
- It can be used in the body of loop or in the body of switch.
- When break encounters loop terminates and control move out of the loop body.

Write a program which asks user to enter an even number, computer will give user at most 3 chances, if user failed to answer in all of the chances, "Game over" message should be displayed on the screen, otherwise "You Win" message should be displayed and game stops immediately.

Write a program to add numbers entered by user. User can enter any number of numbers until he enters 0.

Continue

```
while (condition)
{
    ≡≡≡
    continue;
    ≡≡≡
}
```

- continue is a keyword
- continue can only be used in the body of loop.
- continue transfers the control to the next iteration.