Problem Solving Through Programming in C

Tutorial Session 4

Prof. Anupam Basu
Dept. of Computer Science & Engg.
IIT Kharagpur

Siddhant Mohapatra
PMRF Scholar
IIT Madras

Loop statements

for loop
while loop
do-while loop (execute at least once)

Q. What will be the output?

```
#include <stdio.h>
int main()
{
   float k = 0;
   for (k = 0.5; k < 3; k++)
      printf("I love C\n");
   return 0;
}
```

for (i=0; i<=N; i++)
initialization ↓ condition → decrement or increment

1st iter k=0.5 printed
2nd iter k=1.5 printed
3rd iter k=2.5 printed
k=3.5 loop breaks

a) Error
b) I love C – printed 6 times
c) I love C – printed 5 times
d) I love C – printed 3 times

What is the output of the following code?

```
#include <stdio.h>
int main()
{
  int i=1;
  do
  {
   printf("while vs do-while\n");
  }while(i==1);
  printf("Out of loop");
   return 0;
}
```

No increment/decrement of iterator i

always TRUE

a) 'while vs do-while' --once
b) 'Out of loop' –infinite times
c) Both 'while vs do-while' and 'Out of loop' --once
d) 'while vs do-while' –infinite times

Loop statements

What will be printed when the following code is executed?

```
#include<stdio.h>
int main()
{
   int i=0;
   for(;i<=9;)
   {
        i++;
        printf("%d ", i);
        }
        return 0;
}
```

a) 0 1 2 …9
b) 0 1 2 …10
c) 1 2 3 …9
d) 1 2 3 …10

What is the output of the following code?

```
#include <stdio.h>
int main()
{
  int i=1;
  do
  {
   printf("while vs do-while\n");
  }while(i==1);
  printf("Out of loop");
   return 0;
}
```

a) 'while vs do-while' --once
b) 'Out of loop' –infinite times
c) Both 'while vs do-while' and 'Out of loop' --once
d) 'while vs do-while' –infinite times

Loop statements

Q. What will be the output?

```
#include <stdio.h>
int main()
{
int n;
for(n=10; n<10; n--)
  printf(" %d", n);
return 0;
}
```

a) 10 9 8 7 6 5 4 3 2 1
b) 1 2 3 4 5 6 7 8 9 10
c) No output
d) None of the above

General for loop
i) the conditional is checked at beginning of iteration
ii) decrement/increment occurs at end of iteration

n = 10
n < 10 false loop will not execute

Q. What will be the output?

```
#include<stdio.h>
int main()
{
int k,j;
   for (k=1,j=3;k<=3,j>=1;k++,j--)
   {
        printf("%d,%d\n",k,j);

   }
return 0;
}
```

Logical AND

a) 1,3
b) 1,3
   3,1
c) 1,3
   2,2
   3,1
d) 0,0

k=1, j=3
1st iter
k=2, j=2
2nd iter
k=3, j=1
3rd iter
k=4, j=0

1,3
2,2
3,1

k<=3, j>=0

1,3
2,2
3,1

Flow control statements

break : gets you out of the nearest loop

continue : skip all subsequent statements in the loop and go to next iteration

Q. What will be the output?

```
#include <stdio.h>
int main()
{
int i=0;
for(;;)
   {
      if(i==10)
         break;
    printf("%d ", ++i);
   }
   return 0;
}
```

a) Syntax error
b) 0 1 2 3 4 5 6 7 8 9 10
c) 1 2 3 4 5 6 7 8 9 10
d) 0 1 2 3 4 5 6 7 8 9

1st iter i=1
2nd iter i=2
3rd iter i=3
prefix increment

1, 2, 3, ... 9, 10

10th iter i=10
11th iter i==10 true break

```
for(;;)
{
printf("I");
}
```
infinite loop

Q. What will be the output?

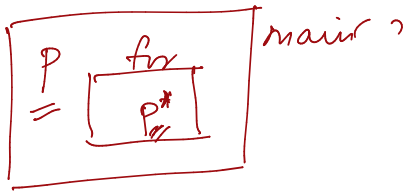
```
#include <stdio.h>
int main()
{
int x=1;
do
   {
   x++;
   continue;
   printf("%d", x);
   break;
}while(x<=2);
printf("\nAfter loop x=%d", x);
printf("\n");
   return 0;
}
```

do { statements; } while(condition);

ignored

a) After loop x=1
b) 1
   After loop x=2
c) 12
   After loop x=3
d) After loop x=3

1st iter x=1+1=2 x<=2 true
2nd iter x=2+1=3 x<=2 false



Miscellaneous exercises

Q. What will be the output of the program?

```
#include <stdio.h>
 int main()
{
int p;
   for (p=0; p<3; p++)
   {
      int p=4;
      printf ("%d,", p);
   }
return 0;
}
```

a) 4,
b) 4,4,4,
c) 4,4,4,4,
d) No output

1st iter
p=0
p++ ⇒ p=1
p<3 true
4, 4, 4,
2nd iter
int p=4
p++ ⇒ p=2
p<3 true
3rd iter
p++ ⇒ p=3
p<3 false

scope of int p=4; inside the for loop.

Q. What should be in the place of ****** so that except i=8, rest of the values of i (as defined in the 'for' loop: i=0 to i=19) will be printed?

```
#include <stdio.h>
 int main()
{
  int i = 0;
   for (i = 0;i< 20; i++)
   {
      if(i==8)
       {
       **********;
        }
      printf("i=%d\n",i);
   }
  return 0;
  }
```

a) break
b) continue
c) switch
d) exit

Miscellaneous exercises

What is the output of the below C program?

```
#include <stdio.h>
int main()
{
short int k=1, j=1;
   while (k <= 4 || j <= 3)
   {
                k=k+2;
                j+=1;
   }
printf("%d,%d", k,j);
return 0;
}
```

a) 5, 4
b) 7, 4
c) 5, 6
d) 6, 4

What will be the value of 'i' after the execution of the program below

```
#include <stdio.h>
int main()
{
int i=1, j;
  for(j=0; j<=10; j+=i)
  {
      i=i+j;
  }
return 0;
}
```

a) 10
b) 11
c) 12
d) 13

Miscellaneous exercises

Q. What will be the output?

```
#include <stdio.h>
int main()
{
if((0 && 1)||(1 && -1))
printf("Condition is true.");
else
printf("Condition is false.");
return 0;
}
```

a) Condition is true
b) Condition is false
c) Compilation error
d) No output

Q. What will be the output?

```
#include <stdio.h>
int main()
{
   switch(printf("IIT"))
   {
      default:
               printf(" Guwahati");
      case 1: printf(" Delhi");
               break;
      case 2: printf(" Kharagpur");
               break;
      case 3: printf(" Madras");
               break;
   }
   return 0;
}
```

a) IIT Guwahati
b) IIT Delhi
c) IIT Kharagpur
d) IIT Madras

Miscellaneous exercises

Q. For the C program given below, if the input given by the user is 7. What will be shown on the output window?

```
#include<stdio.h>
int main()
{
int n,i=2;
scanf("%d",&n);
  do
  {
if(n%i==0)
{
  printf("The number is odd");
}
i++;
}
while(i<n);
printf("The number is prime");
   return 0;
}
```

n = 7

1st iter
7%2 == 1 == 0 false
i = 3
3 < 7 true

2nd iter
7%3 == 0 false
1
i = 4
4 < 7 true

3rd iter
7%4 == 0 false
3
i = 5
5 < 7 true

4th iter
7%5 == 0 false
2
i = 6
6 < 7 true

5th iter
7%6 == 0 false
1
i = 7
7 < 7 ⇒ false.
loop stops

a) The number is odd
b) The number is prime
c) The number is odd The number is prime
d) Syntax error

Miscellaneous exercises

371

$3^3+7^3+1^3 = 371$

Write a code to enter a 3-digit integer and check if Armstrong number (sum of digits raised to the power of total no. of digits is equal to the number).

```
#include <stdio.h>
#include <math.h>
int main()   {
    int n, m, sum = 0, x;
    printf("Enter a 3 digit positive integer: ");
    scanf("%d", &n);
    m = n;
    while (n > 0)     {
        x = n%10;
        sum = sum + pow(x,3);
        n = n/10;     }
    if (m == sum)  printf("%d is an Armstrong number.\n", m);
    else    printf("%d is not an Armstrong number.\n", m);
   return 0;
}
```

```
sum=0.0; m=n
while(m>0)
{
   x = m%10;
   sum += x*x*x;
   m = (m-x)/10;
}
if(sum==n) armstrong no.
```

1st iter
x = 1
sum = 1
m = 37

2nd iter
x = 7
sum = $1+7^3$
m = 3

3rd iter
x = 3
sum = $1+7^3+3^3$
m = 0