



```
*****  
* convolve.c  
***** /
```

```
/* Standard includes */  
#include <assert.h>  
#include <math.h>  
#include <stdio.h> /* malloc(), realloc() */
```

```
/* Our includes */  
#include "base.h"  
#include "error.h"  
#include "convolve.h"  
#include "klt_util.h" /* printing */
```

```
#define MAX_KERNEL_WIDTH 71
```

```
typedef struct {  
    int width;  
    float data[MAX_KERNEL_WIDTH];  
} ConvolutionKernel;
```

```
/* Kernels */
```

Fundamentals of Programming

Lecture 7

Loops

Loops



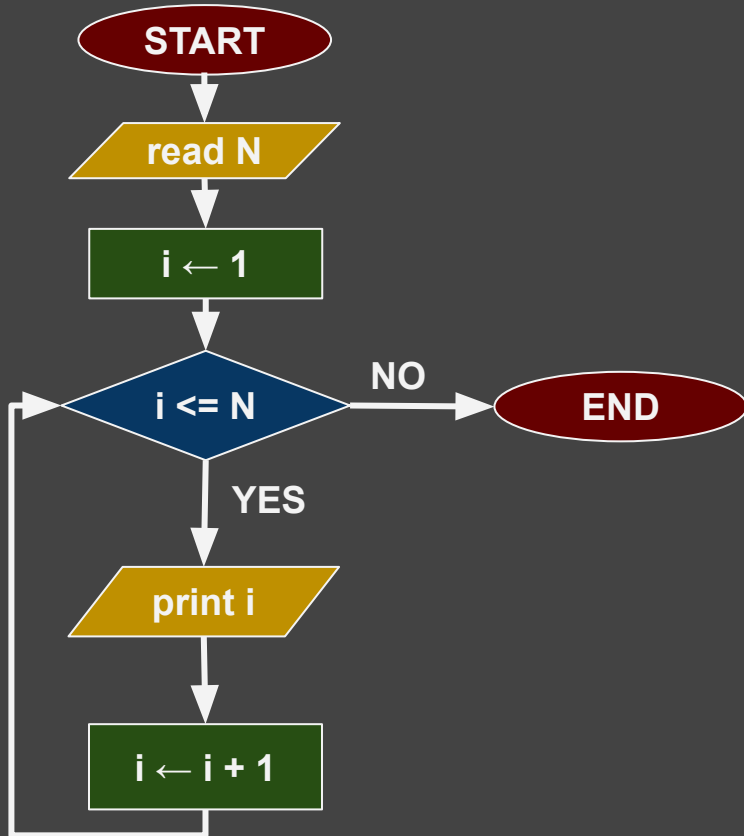
K. N. Toosi
University of Technology



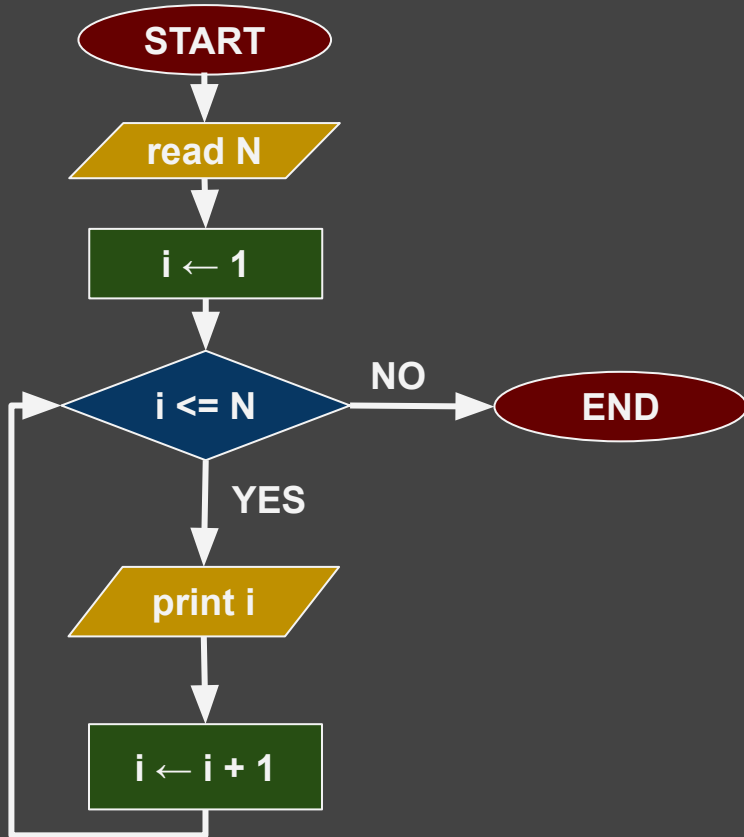
Loops



K. N. Toosi
University of Technology

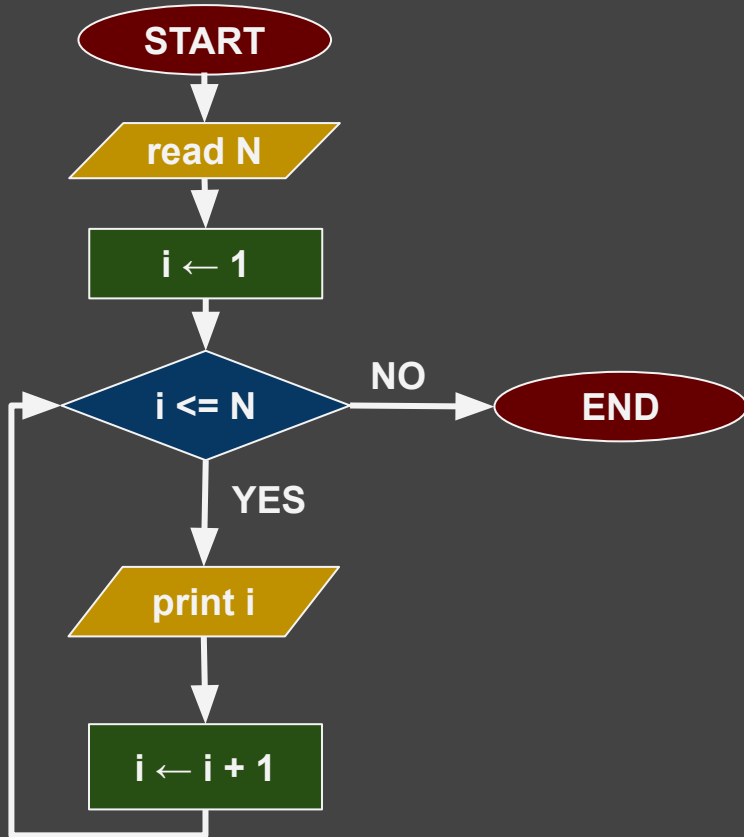


Loops



```
goto label;  
... ..  
... ..  
label:  
... ..  
... ..
```

Loops



```
#include <stdio.h>
int main() {
    int N,i;
    scanf("%d", &N);

    i = 1;

myloop:
    if (i > N) {
        goto endloop;
    }
    printf("%d\n", i);

    i = i+1;

    goto myloop;
endloop:
    return 0;
}
```

loop1.c



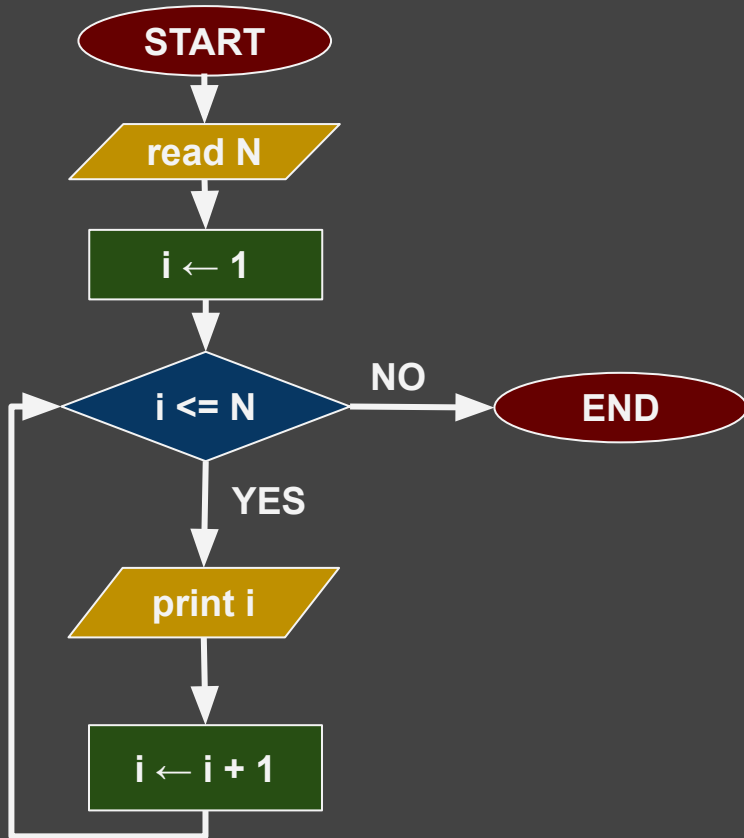
K. N. Toosi
University of Technology



K. N. Toosi
University of Technology

Avoid using the goto command!

The while loop



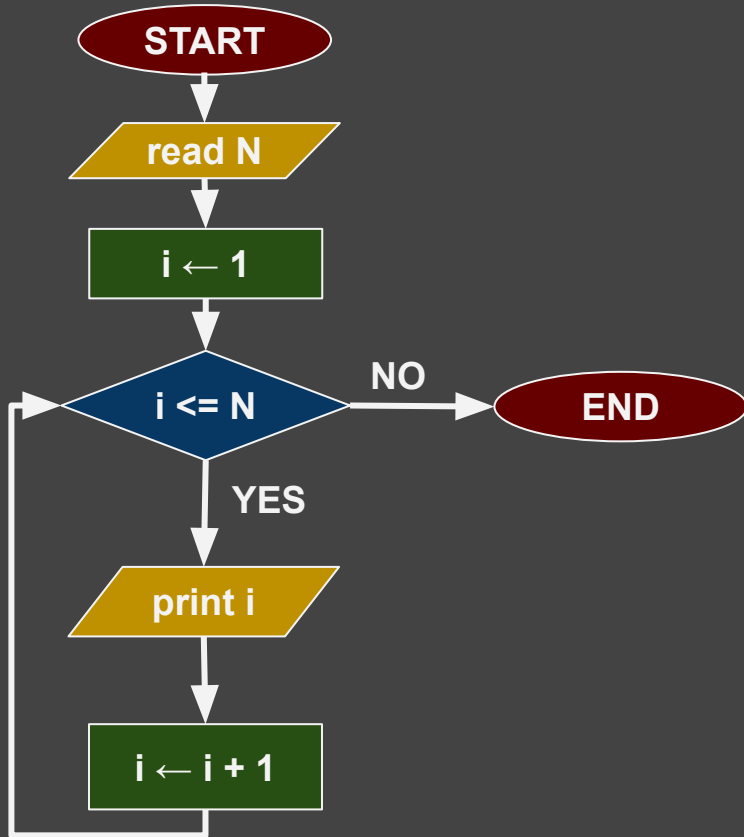
```
if (CONDITION) {
```

```
}
```

```
while (CONDITION) {
```

```
}
```

Loops



```
#include <stdio.h>
int main() {
    int N,i;

    scanf("%d", &N);

    i = 1;

    while (i <= N) {

        printf("%d\n", i);
        i = i+1;

    }

    return 0;
}
```

loop2.c



K. N. Toosi
University of Technology

Loops



loop3.c

```
int N,i,p;

scanf("%d", &N);

i = 0;
p = 1;

while (i < N) {

    p = p * 2;
    i = i + 1;

    printf("%d: %d\n", i, p);

}
```

Loops



loop3.c

```
int N,i,p;

scanf("%d", &N);

i = 0;
p = 1;

while (i < N) {

    p = p * 2;
    i = i + 1;

    printf("%d: %d\n", i, p);

}
```

```
CS@kntu:lecture7$ gcc loop3.c -o loop3 && ./loop3
20
1: 2
2: 4
3: 8
4: 16
5: 32
6: 64
7: 128
8: 256
9: 512
10: 1024
11: 2048
12: 4096
13: 8192
14: 16384
15: 32768
16: 65536
17: 131072
18: 262144
19: 524288
20: 1048576
```

Loops



loop3.c

```
int N,i,p;

scanf("%d", &N);

i = 0;
p = 1;

while (i < N) {

    p = p * 2;
    i = i + 1;

    printf("%d: %d\n", i, p);

}
```

loop4.c

```
int N,i,p;

scanf("%d", &N);

i = 0;
p = 1;

while (i < N) {

    p *= 2;
    i += 1;

    printf("%d: %d\n", i, p);

}
```

Loops



loop3.c

```
int N,i,p;

scanf("%d", &N);

i = 0;
p = 1;

while (i < N) {

    p = p * 2;
    i = i + 1;

    printf("%d: %d\n", i, p);

}
```

loop4.c

```
int N,i,p;

scanf("%d", &N);

i = 0;
p = 1;

while (i < N) {

    p *= 2;
    i += 1;

    printf("%d: %d\n", i, p);

}
```

loop5.c

```
int N,i,p;

scanf("%d", &N);

i = 0;
p = 1;

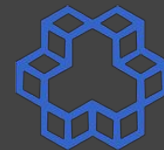
while (i < N) {

    p *= 2;
    ++i;

    printf("%d: %d\n", i, p);

}
```

Loops



loop4.c

```
int N,i,p;

scanf("%d", &N);

i = 0;
p = 1;

while (i < N) {

    p *= 2;
    i += 1;

    printf("%d: %d\n", i, p);

}
```

loop5.c

```
int N,i,p;

scanf("%d", &N);

i = 0;
p = 1;

while (i < N) {

    p *= 2;
    ++i;

    printf("%d: %d\n", i, p);

}
```

loop6.c

```
int N,i,p;

scanf("%d", &N);

i = 0;
p = 1;

while (i < N) {

    p *= 2;
    i++;

    printf("%d: %d\n", i, p);

}
```

Loops



K. N. Toosi
University of Technology

loop7.c

```
int N,i,p;

scanf("%d", &N);

i = 0;
p = 1;

while (i < N) {

    p *= 2;
    i++;

    printf("2^%d = %d\n", i, p);

}
```

Loops



loop7.c

```
int N,i,p;

scanf("%d", &N);

i = 0;
p = 1;

while (i < N) {

    p *= 2;
    i++;

    printf("2^%d = %d\n", i, p);

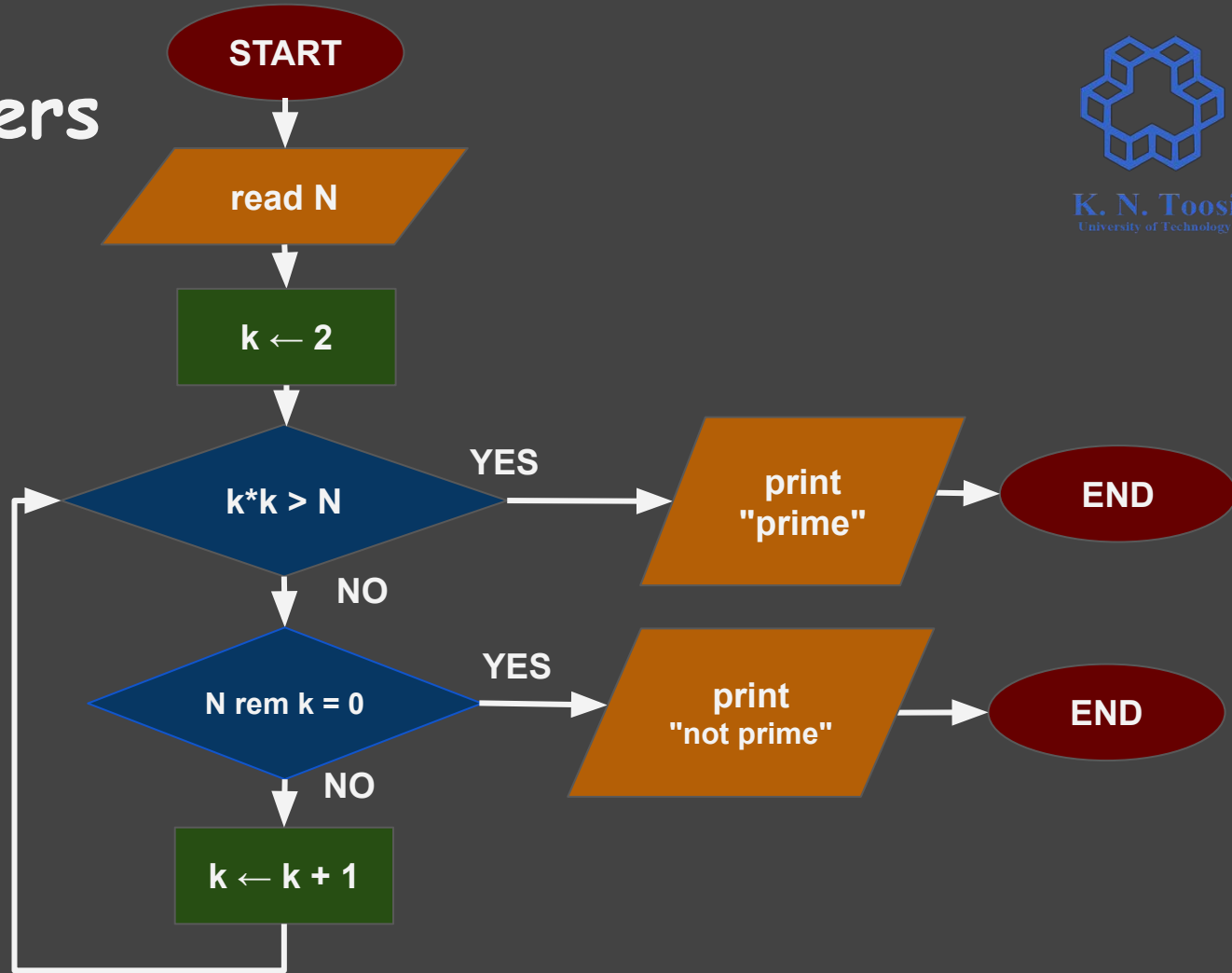
}
```

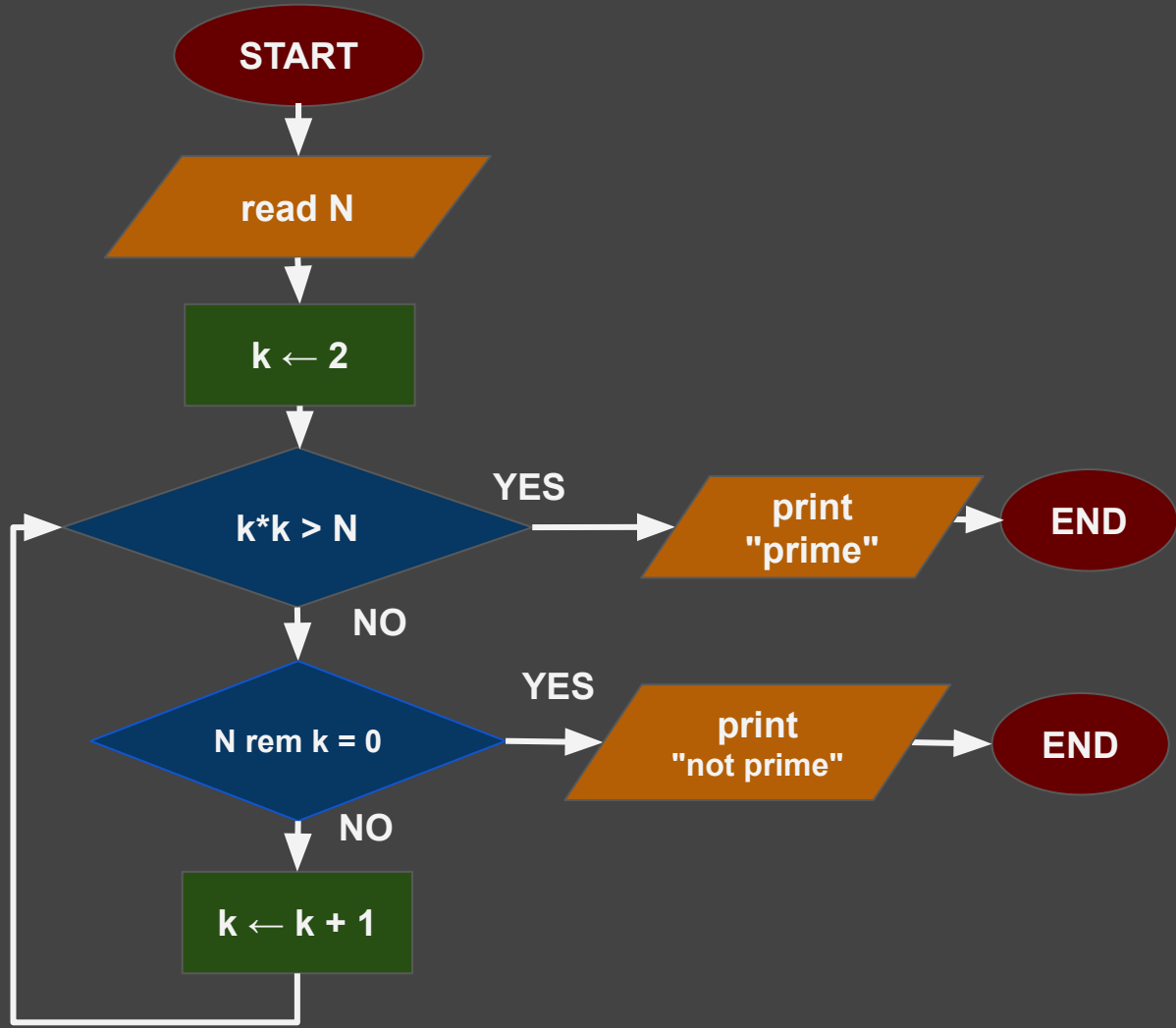
```
CS@kntu:lecture7$ gcc loop7.c -o loop7 && ./loop7
20
2^1 = 2
2^2 = 4
2^3 = 8
2^4 = 16
2^5 = 32
2^6 = 64
2^7 = 128
2^8 = 256
2^9 = 512
2^10 = 1024
2^11 = 2048
2^12 = 4096
2^13 = 8192
2^14 = 16384
2^15 = 32768
2^16 = 65536
2^17 = 131072
2^18 = 262144
2^19 = 524288
2^20 = 1048576
CS@kntu:lecture7$
```

Prime numbers



K. N. Toosi
University of Technology





```

int main() {
    int N,k;
    scanf("%d", &N);
    k = 2;

loop1:
    if (k*k > N) {
        printf("Prime!\n");
        goto endprog;
    }

    if (N % k == 0) {
        printf("Not Prime!\n");
        goto endprog;
    }

    k = k + 1;

    goto loop1;

endprog:

    return 0;
}

```

prime0.c



K. N. Toosi
University of Technology

Avoid using the goto command!

Homework/Exam codes with a goto command are not graded!

```
int main() {
    int N,k;
    scanf("%d", &N);
    k = 2;

loop1:
    if (k*k > N) {
        printf("Prime!\n");
        goto endprog;
    }

    if (N % k == 0) {
        printf("Not Prime!\n");
        goto endprog;
    }

    k = k + 1;

    goto loop1;

endprog:

    return 0;
}
```

prime0.c

```
int main() {
    int N,k;
    scanf("%d", &N);

    k = 2;

    while (k * k <= N  && N % k != 0) {
        k = k + 1;
    }

    if (k * k <= N) {
        printf("Not Prime!\n");
    }
    else {
        printf("Prime!\n");
    }

    return 0;
}
```

prime1.c



```

int main() {
    int N,k;
    scanf("%d", &N);

    k = 2;

    while (k * k <= N && N % k != 0) {
        k = k + 1;
    }

    if (k * k <= N) {
        printf("Not Prime!\n");
    }
    else {
        printf("Prime!\n");
    }

    return 0;
}

```

prime1.c

```

int main() {
    int N,k, prime;
    scanf("%d", &N);

    prime = 1;
    k = 2;
    while (k*k <= N) {

        if (N % k == 0) {
            prime = 0;
        }
        k = k + 1;
    }

    if (prime == 1)
        puts("Prime");
    else
        puts("Not Prime");

    return 0;
}

```

prime2.c



K. N. Toosi
University of Technology

prime2.c

```
int main() {
    int N,k, prime;
    scanf("%d", &N);

    prime = 1;
    k = 2;
    while (k*k <= N) {

        if (N % k == 0) {
            prime = 0;
        }
        k = k + 1;
    }

    if (prime == 1)
        puts("Prime");
    else
        puts("Not Prime");

    return 0;
}
```

prime2.5.c

```
int main() {
    int N,k, prime;
    scanf("%d", &N);

    prime = 1;
    k = 2;
    while (k*k <= N) {

        if (N % k == 0)
            prime = 0;

        k = k + 1;
    }

    if (prime == 1)
        puts("Prime");
    else
        puts("Not Prime");

    return 0;
}
```



```

int main() {
    int N,k, prime;
    scanf("%d", &N);

    prime = 1;
    k = 2;
    while (k*k <= N) {

        if (N % k == 0) {
            prime = 0;
        }
        k = k + 1;
    }

    if (prime == 1)
        puts("Prime");
    else
        puts("Not Prime");

    return 0;
}

```

prime2.c

```

int main() {
    int N,k, prime;
    scanf("%d", &N);

    prime = 1;
    k = 2;
    while (k*k <= N) {

        if (N % k == 0) {
            prime = 0;
            break;
        }
        k = k + 1;
    }

    if (prime == 1)
        puts("Prime");
    else
        puts("Not Prime");

    return 0;
}

```

prime3.c



K. N. Toosi
University of Technology

```

int main() {
    int N,k, prime;
    scanf("%d", &N);

    prime = 1;
    k = 2;
    while (k*k <= N) {

        if (N % k == 0) {
            prime = 0;
        }
        k = k + 1;
    }

    if (prime == 1)
        puts("Prime");
    else
        puts("Not Prime");

    return 0;
}

```

prime2.c

```

int main() {
    int N,k, prime;
    scanf("%d", &N);

    prime = 1;
    k = 2;
    while (k*k <= N) {

        if (N % k == 0) {
            prime = 0;
            break;
        }
        k = k + 1;
    }

    if (prime == 1)
        puts("Prime");
    else
        puts("Not Prime");

    return 0;
}

```

prime3.c



K. N. Toosi
University of Technology

C Keywords (reserved words)

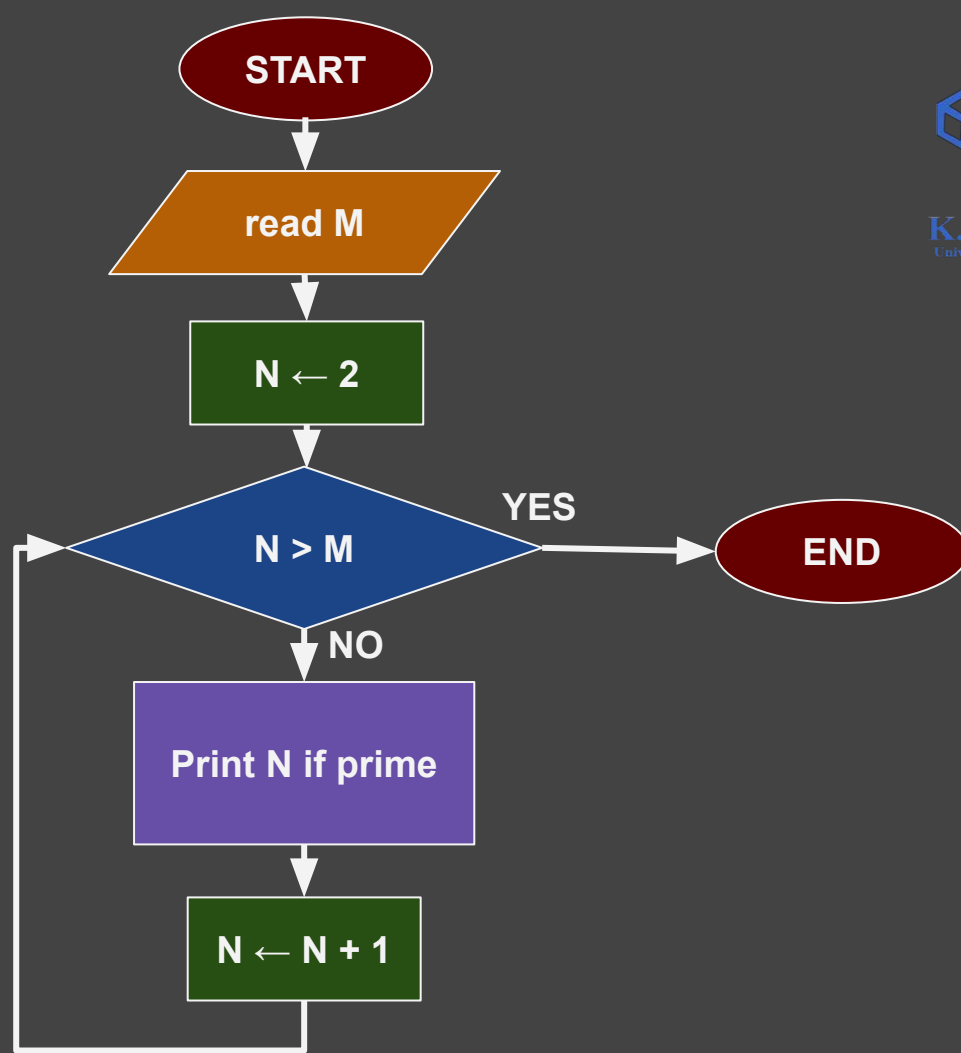


K. N. Toosi
University of Technology

Keywords in C Programming			
auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

<https://www.programiz.com/c-programming/list-all-keywords-c-language>

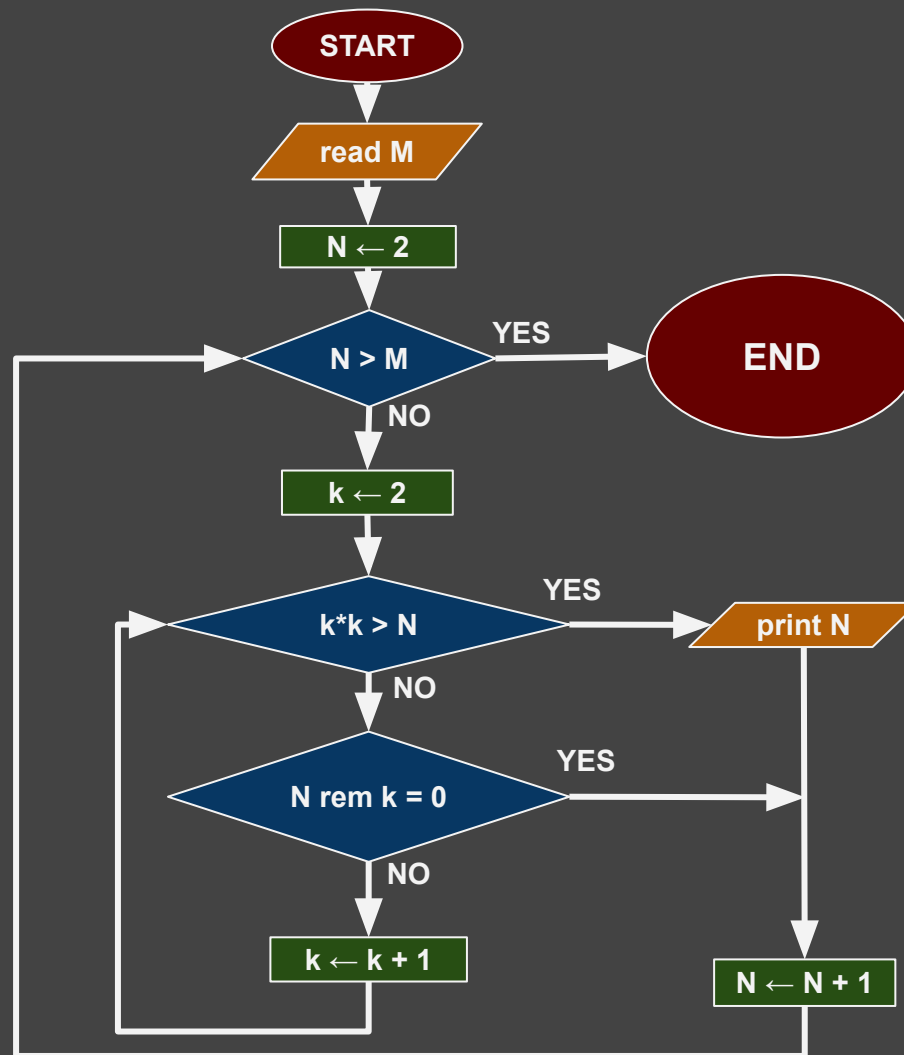
Prime numbers

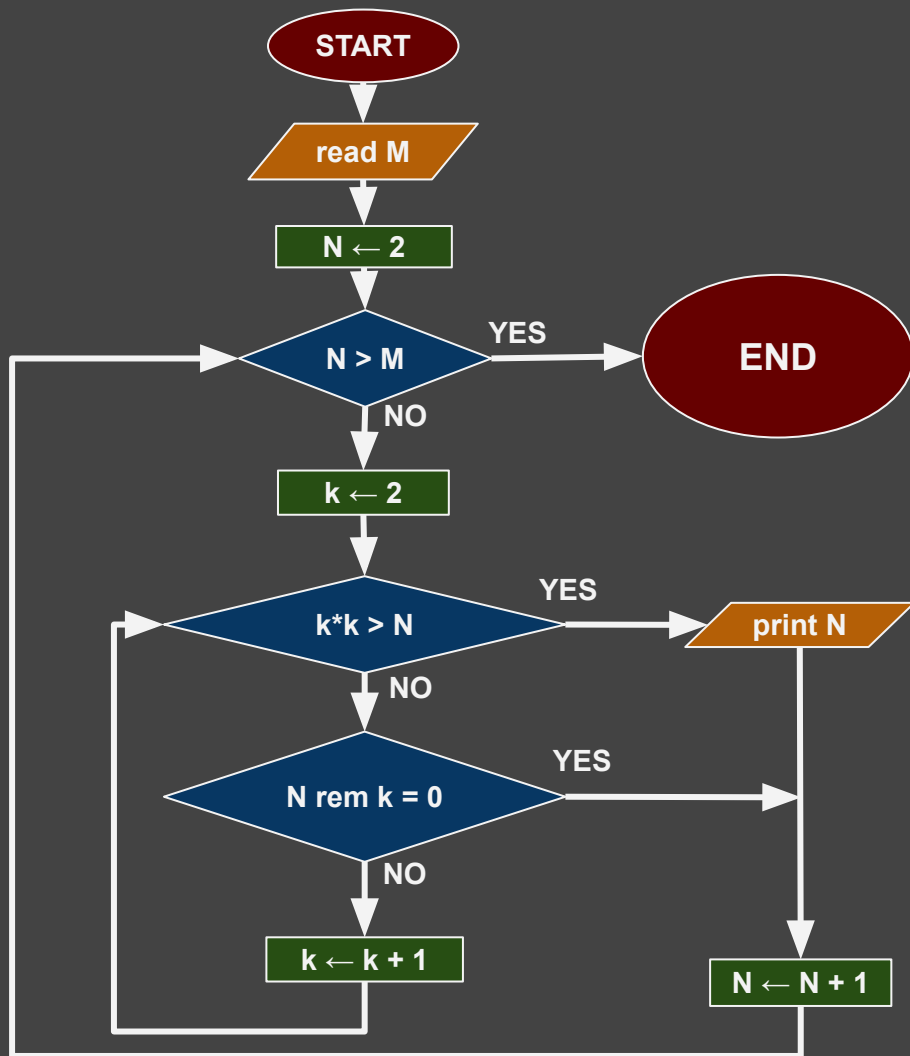


Prime numbers



K. N. Toosi
University of Technology





prime3.c

```

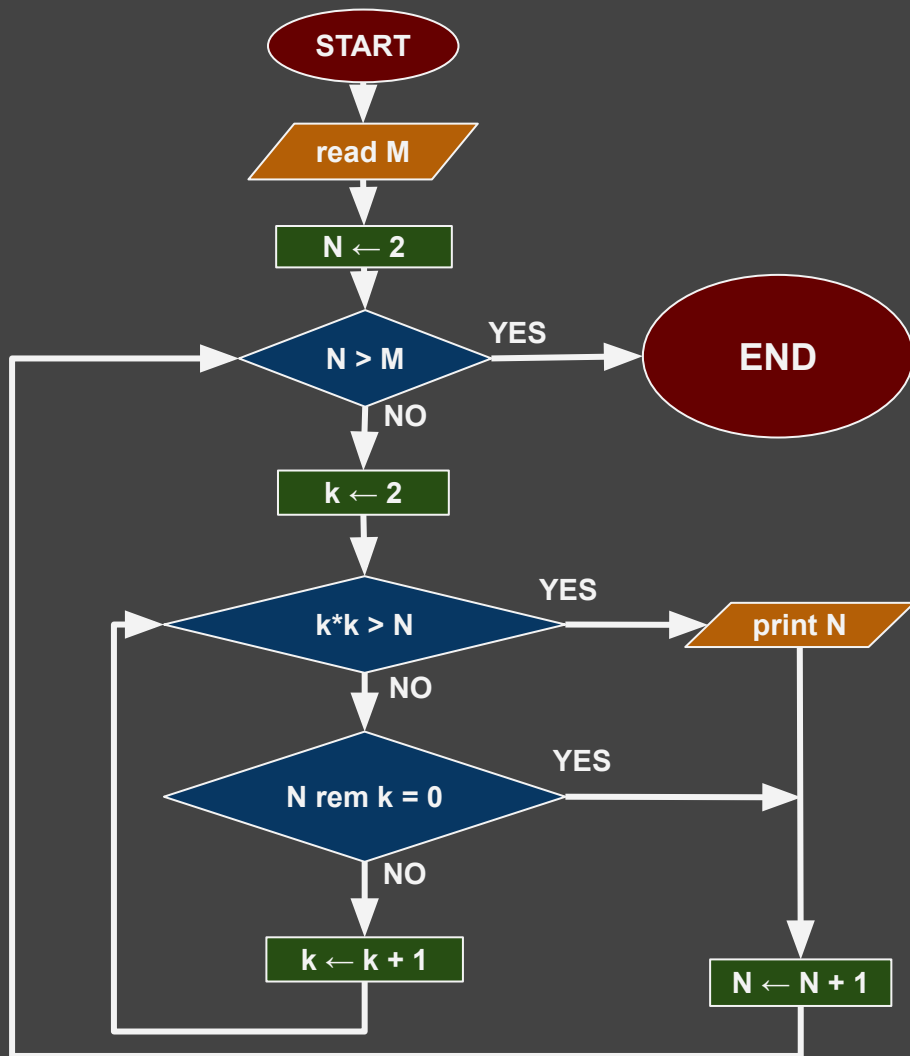
int M, N, k, prime;
scanf("%d", &M);
N = 2;
while (N <= M) {

```

```

    N = N + 1;
}

```

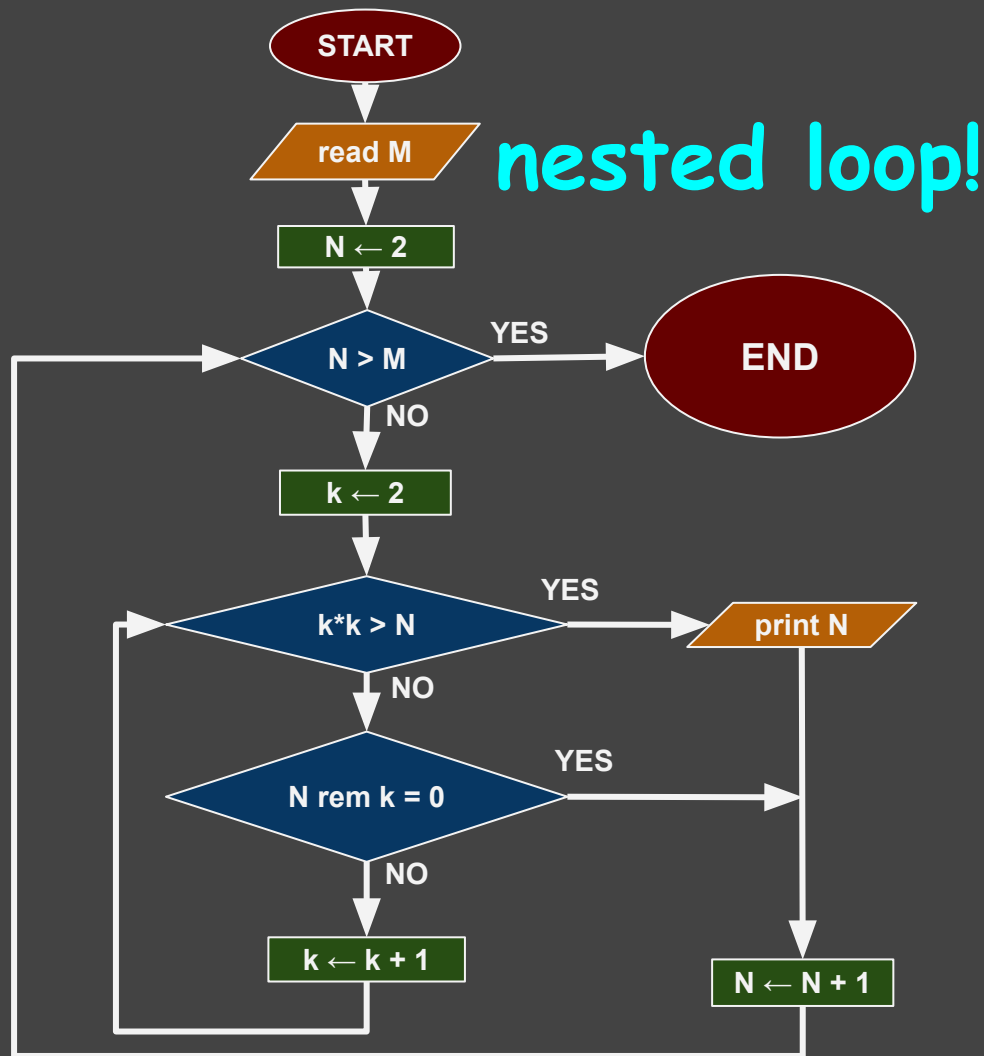


```

int M, N, k, prime;
scanf("%d", &M);
N = 2;
while (N <= M) {
    prime = 1;
    k = 2;
    while (k*k <= N) {
        if (N % k == 0) {
            prime = 0;
            break;
        }
        k = k + 1;
    }
    if (prime == 1)
        printf("%d\n", N);
    N = N + 1;
}

```

prime3.c



```

int M, N, k, prime;
scanf("%d", &M);
N = 2;
while (N <= M) {
    prime = 1;
    k = 2;
    while (k*k <= N) {
        if (N % k == 0) {
            prime = 0;
            break;
        }
        k = k + 1;
    }
    if (prime == 1)
        printf("%d\n", N);
    N = N + 1;
}
  
```

prime3.c

Counter-controlled iterations



K. N. Toosi
University of Technology

```
k = 1;  
while (k <= N) {  
  
    k++;  
  
}
```



Counter-controlled iterations



K. N. Toosi
University of Technology

```
k = 0;  
while (k < N) {  
  
    k++;  
  
}
```



Counter-controlled iterations



K. N. Toosi
University of Technology



```
k = 1;  
while (k <= N) {  
  
    k++;  
}
```



```
k = 0;  
while (k < N) {  
  
    k++;  
}
```


Counter-controlled iterations



Adding up n numbers:

Write a program that reads students' scores in a class and prints their total sum.

Counter-controlled iterations



K. N. Toosi
University of Technology

Adding up n numbers:

Write a program that reads students' scores in a class and prints their sum.

```
float a, sum;
int n, k;

printf("Enter no. of students: ");
scanf("%d", &n);

sum = 0;
k = 1;
while (k <= n) {
    scanf("%f", &a);
    sum = sum + a;
    k++;
}
printf("sum=%f\n", sum);
```

sumscores.c

Counter-controlled iterations



K. N. Toosi
University of Technology

Adding up n numbers:

Write a program that reads students' scores in a class and prints their sum.

```
CS@kntu:lecture7$ gcc sumscores.c && ./a.out
Enter no. of students: 7
12.5
20
17.45
14.0
12.25
7.5
19.99
sum=103.689995
CS@kntu:lecture7$
```

```
float a,sum;
int n,k;

printf("Enter no. of students: ");
scanf("%d", &n);

sum = 0;
k = 1;
while (k <= n) {
    scanf("%f", &a);
    sum = sum + a;
    k++;
}
printf("sum=%f\n", sum);
```

sumscores.c

Counter-controlled iterations



K. N. Toosi
University of Technology

```
float a,sum;
int n,k;

printf("Enter no. of students: ");
scanf("%d", &n);

sum = 0;
k = 1;
while (k <= n) {
    scanf("%f", &a);
    sum = sum + a;
    k++;
}
printf("sum=%f\n", sum);
```

sumscores.c

```
float a,sum;
int n,k;

printf("Enter no. of students: ");
scanf("%d", &n);

sum = 0;
k = 1;
while (k <= n) {
    scanf("%f", &a);
    sum = sum + a;
    k++;
}
printf("res=%f\n", sum/n);
```

Counter-controlled iterations



K. N. Toosi
University of Technology

```
float a, sum;
int n, k;

printf("Enter no. of students: ");
scanf("%d", &n);

sum = 0;
k = 1;
while (k <= n) {
    scanf("%f", &a);
    sum = sum + a;
    k++;
}
printf("sum=%f\n", sum);
```

sumscores.c

```
float a, sum;
int n, k;

printf("Enter no. of students: ");
scanf("%d", &n);

sum = 0;
k = 1;
while (k <= n) {
    scanf("%f", &a);
    sum = sum + a;
    k++;
}
printf("average=%f\n", sum/n);
```

averagescores.c

Sentinel-controlled iterations



K. N. Toosi
University of Technology

```
float a, sum;
```

sentinel1.c

```
sum = 0;
```

```
while (1 == 1) {  
    scanf("%f", &a);
```

```
    if (a < 0)  
        break;
```

```
    sum = sum + a;
```

```
}
```

```
printf("sum=%f\n", sum);
```



Sentinel-controlled iterations



K. N. Toosi
University of Technology

```
float a, sum;
```

sentinel1.c

```
sum = 0;
```

```
while (1 == 1) {
```

```
    scanf("%f", &a);
```

```
    if (a < 0)
```

```
        break;
```

```
    sum = sum + a;
```

```
}
```

```
printf("sum=%f\n", sum);
```



Sentinel-controlled iterations



K. N. Toosi
University of Technology

```
float a, sum;
```

sentinel1.c

```
sum = 0;
```

```
while (1 == 1) {  
    scanf("%f", &a);
```

```
    if (a < 0)  
        break;
```

```
    sum = sum + a;
```

```
}
```

```
printf("sum=%f\n", sum);
```



Sentinel-controlled iterations



K. N. Toosi
University of Technology

```
float a, sum;

sum = 0;
while (1 == 1) {
    scanf("%f", &a);

    if (a < 0)
        break;

    sum = sum + a;
}

printf("sum=%f\n", sum);
```

sentinel1.c

```
CS@kntu:lecture7$ gcc sentinel1.c && ./a.out
12.6
17.8
20
10.5
-1
sum=60.900002
```

Sentinel-controlled iterations



K. N. Toosi
University of Technology

```
float a, sum;
```

sentinel1.c

```
sum = 0;
while (1 == 1) {
    scanf("%f", &a);

    if (a < 0)
        break;

    sum = sum + a;
}

printf("sum=%f\n", sum);
```

```
float a, sum;
```

sentinel2.c

```
sum = 0;
while (1) {
    scanf("%f", &a);

    if (a < 0)
        break;

    sum = sum + a;
}

printf("sum=%f\n", sum);
```

Sentinel-controlled iterations



K. N. Toosi
University of Technology

```
float a, sum;
sum = 0;

while (1) {
    scanf("%f", &a);

    if (a < 0)
        break;

    sum = sum + a;
}

printf("sum=%f\n", sum);
```

sentinel2.c

```
float a, sum;
sum = 0;

scanf("%f", &a);
while (a >= 0) {

    sum = sum + a;

    scanf("%f", &a);
}

printf("sum=%f\n", sum);
```

sentinel3.c

Sentinel-controlled iterations



K. N. Toosi
University of Technology

```
float a, sum;
sum = 0;

while (1) {
    scanf("%f", &a);

    if (a < 0)
        break;

    sum = sum + a;
}

printf("sum=%f\n", sum);
```

sentinel2.c

```
float a, sum;
sum = 0;

scanf("%f", &a);
while (a >= 0) {

    sum = sum + a;

    scanf("%f", &a);
}

printf("sum=%f\n", sum);
```

sentinel3.c

Sentinel-controlled iterations



K. N. Toosi
University of Technology

```
float a, sum;
sum = 0;

while (1) {
    scanf("%f", &a);

    if (a < 0)
        break;

    sum = sum + a;
}

printf("sum=%f\n", sum);
```

sentinel2.c

```
float a, sum;
sum = 0;

scanf("%f", &a);
while (a >= 0) {

    sum = sum + a;

    scanf("%f", &a);
}

printf("sum=%f\n", sum);
```

sentinel3.c

Compute Average?



K. N. Toosi
University of Technology

```
float a, sum;
sum = 0;

scanf("%f", &a);
while (a >= 0) {

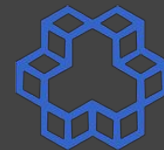
    sum = sum + a;

    scanf("%f", &a);
}

printf("sum=%f\n", sum);
```

sentinel3.c

Compute Average?



K. N. Toosi
University of Technology

```
float a, sum;
sum = 0;

scanf("%f", &a);
while (a >= 0) {

    sum = sum + a;

    scanf("%f", &a);
}

printf("sum=%f\n", sum);
```

sentinel3.c

```
float a, sum, n;
sum = 0;
n = 0;

scanf("%f", &a);
while (a >= 0) {

    sum = sum + a;
    n++;

    scanf("%f", &a);
}

printf("sum=%f\n", sum/n);
```

sentinel4.c

Practice: factorial

Write a program that reads a positive number n and prints its factorial $n!$



K. N. Toosi
University of Technology

Practice: factorial



K. N. Toosi
University of Technology

Write a program that reads a positive number n and prints its factorial $n!$

$$n! = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$



Practice: factorial

Write a program that reads a positive number n and prints its factorial $n!$

$$n! = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

```
int n,i;

printf("Enter a positive number: ");
scanf("%d", &n);

i = 1;
while (i <= n) {

    i++;

}
```

factorial1.c



Practice: factorial

Write a program that reads a positive number n and prints its factorial $n!$

$$n! = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

```
int n,i, fact;

printf("Enter a positive number: ");
scanf("%d", &n);

fact = 1;
i = 1;
while (i <= n) {
    fact *= i;
    i++;
}

printf("%d! = %d\n",n, fact);
```

factorial1.c



Practice: factorial

Write a program that reads a positive number n and prints its factorial $n!$

$$n! = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

Keep reading until the user enters a positive number.

```
int n,i, fact;

printf("Enter a positive number: ");
scanf("%d", &n);

fact = 1;
i = 1;
while (i <= n) {
    fact *= i;
    i++;
}

printf("%d! = %d\n",n, fact);
```

factorial1.c

Practice: factorial



K. N. Toosi
University of Technology

Write a program that reads a positive number n and prints its factorial $n!$

$$n! = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

Keep reading until the user enters a positive number.

factorial2.c

```
while (1) {  
    printf("Enter a positive number: ");  
    scanf("%d", &n);  
    if (n > 0)  
        break;  
}  
  
fact = 1;  
i = 1;  
while (i <= n) {  
    fact *= i;  
    i++;  
}  
printf("%d! = %d\n", n, fact);
```



Practice: factorial

Write a program that reads a positive number n and prints its factorial $n!$

$$n! = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

Keep reading until the user enters a positive number.

factorial2.c

```
while (1) {  
    printf("Enter a positive number: ");  
    scanf("%d", &n);  
    if (n > 0)  
        break;  
}
```

```
fact = 1;  
i = 1;
```

```
CS@kntu:lecture7$ gcc factorial2.c && ./a.out  
Enter a positive number: -1  
Enter a positive number: -2  
Enter a positive number: 0  
Enter a positive number: 1  
1! = 1
```



Practice: factorial

Write a program that reads a positive number n and prints its factorial $n!$

$$n! = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

Keep reading until the user enters a positive number.

```
n = -1;
while (n >= 0) {
    printf("Enter a positive number: ");
    scanf("%d", &n);
}

fact = 1;
i = 1;
while (i <= n) {
    fact *= i;
    i++;
}
printf("%d! = %d\n", n, fact);
```

factorial3.c



Practice: factorial

Write a program that reads a positive number n and prints its factorial $n!$

$$n! = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$$

Keep reading until the user enters a positive number.

factorial4.c

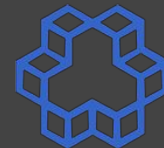
```
do {  
    printf("Enter a positive number: ");  
    scanf("%d", &n);  
} while (n >= 0);  
  
fact = 1;  
i = 1;  
while (i <= n) {  
    fact *= i;  
    i++;  
}  
printf("%d! = %d\n", n, fact);
```


Do-while loop



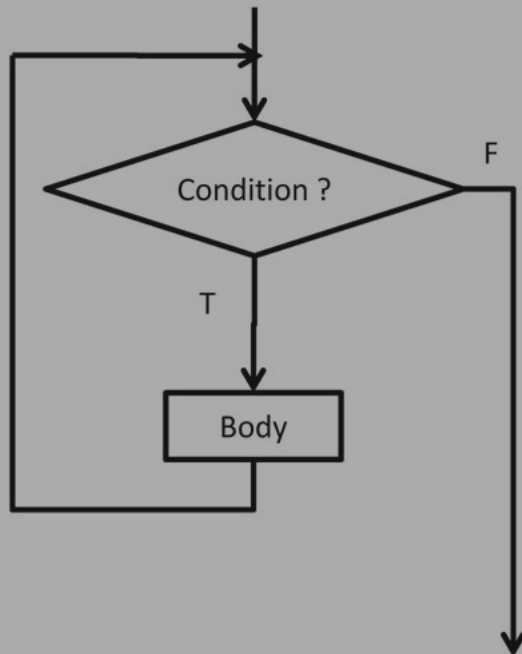
K. N. Toosi
University of Technology

```
do {  
  
} while (CONDITION) ;
```

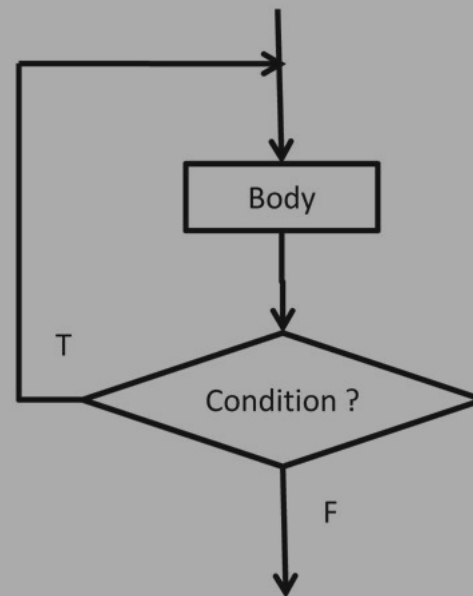


While versus Do-While Loops

```
while( condition )  
body;
```



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



Be careful about do-while loops!



K. N. Toosi
University of Technology



Example:



```
int N = 10;
int k;

k = 1;
while (k <= N) {

    printf("%d\n",k*k);

    k++;
}
```

print_squares.c

Example:



```
int N = 10;
int k;

k = 1;
while (k <= N) {

    printf("%d\n",k*k);

    k++;
}
```

print_squares.c

The for loop



```
int N = 10;      print_squares.c
int k;

k = 1;
while (k <= N) {

    printf("%d\n", k*k);

    k++;
}
}
```

```
int N = 10;      print_squares_for.c
int k;

for (k = 1; k <= N; k++) {

    printf("%d\n", k*k);

}
}
```

The for loop



K. N. Toosi
University of Technology

```
STATEMENT1;  
while (CONDITION) {  
  
    // body of loop  
  
    STATEMENT2;  
}
```

```
for (STATEMENT1; CONDITION; STATEMENT2) {  
  
    // body of loop  
  
}
```

The for loop



K. N. Toosi
University of Technology

```
STATEMENT1;  
while (CONDITION) {  
  
    // body of loop  
  
    STATEMENT2;  
}
```

```
for (STATEMENT1; CONDITION; STATEMENT2) {  
  
    // body of loop  
  
}
```


Practice:

Write the factorial program using a **for** loop.



K. N. Toosi
University of Technology

Practice:



K. N. Toosi
University of Technology

```
for (i = 0, j = N; i <= N; i++,j--) {  
    printf("%d,%d\n",i,j);  
}
```