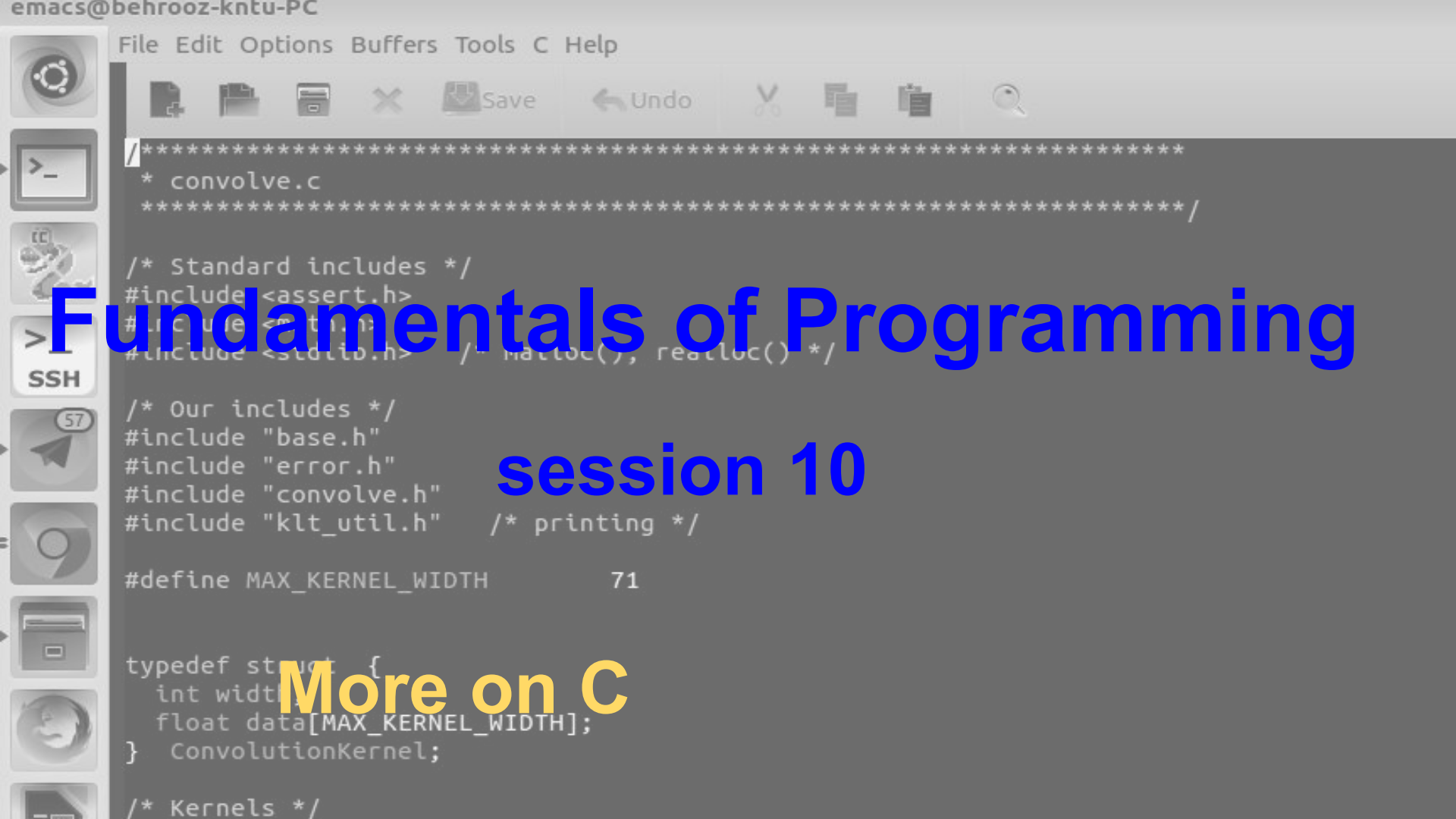

emacs@behrooz-kntu-PC
File Edit Options Buffers Tools C Help
Save
Undo
/**********
* convolve.c
**********/
/* Standard includes */
#include <assert.h>
/* Our includes */
#include "base.h"
#include "error.h"
#include "convolve.h"
#include "klt_util.h" /* printing */
#define MAX_KERNEL_WIDTH 71
typedef struct {
int width;
float data[MAX_KERNEL_WIDTH];
} ConvolutionKernel;
/* Kernels */
SSH
Fundamentals of Programming
session 10
More on C

printing prompts

```c
#include <stdio.h>
#include <time.h>


int main() {
  int a,b;

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

  printf("Enter b: ");
  scanf("%d", &b);

  printf("%d + %d = %d\n", a, b, a+b);


  return 0;
}
```

printing prompts

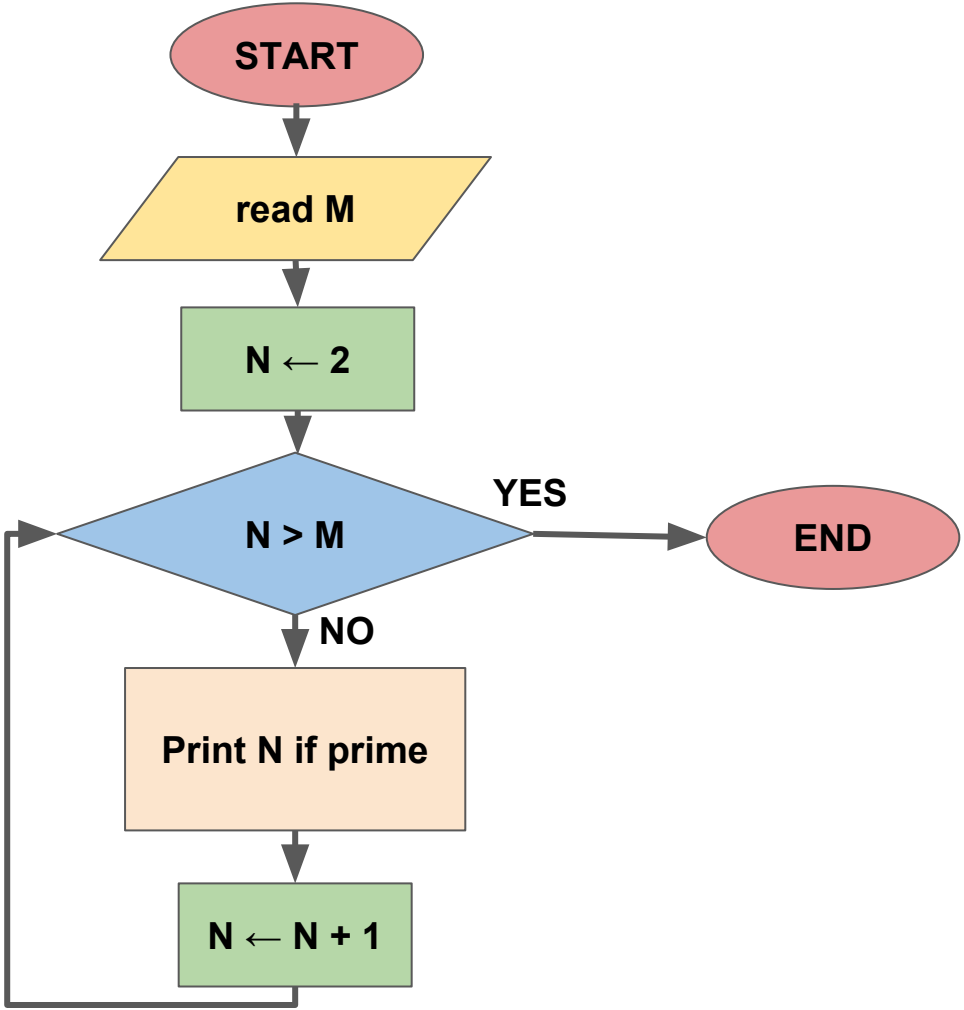
```c
#include <stdio.h>
#include <time.h>


int main() {
  int a,b;

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

  printf("Enter b: ");
  scanf("%d", &b);

  printf("%d + %d = %d\n", a, b, a+b);


  return 0;
}
```

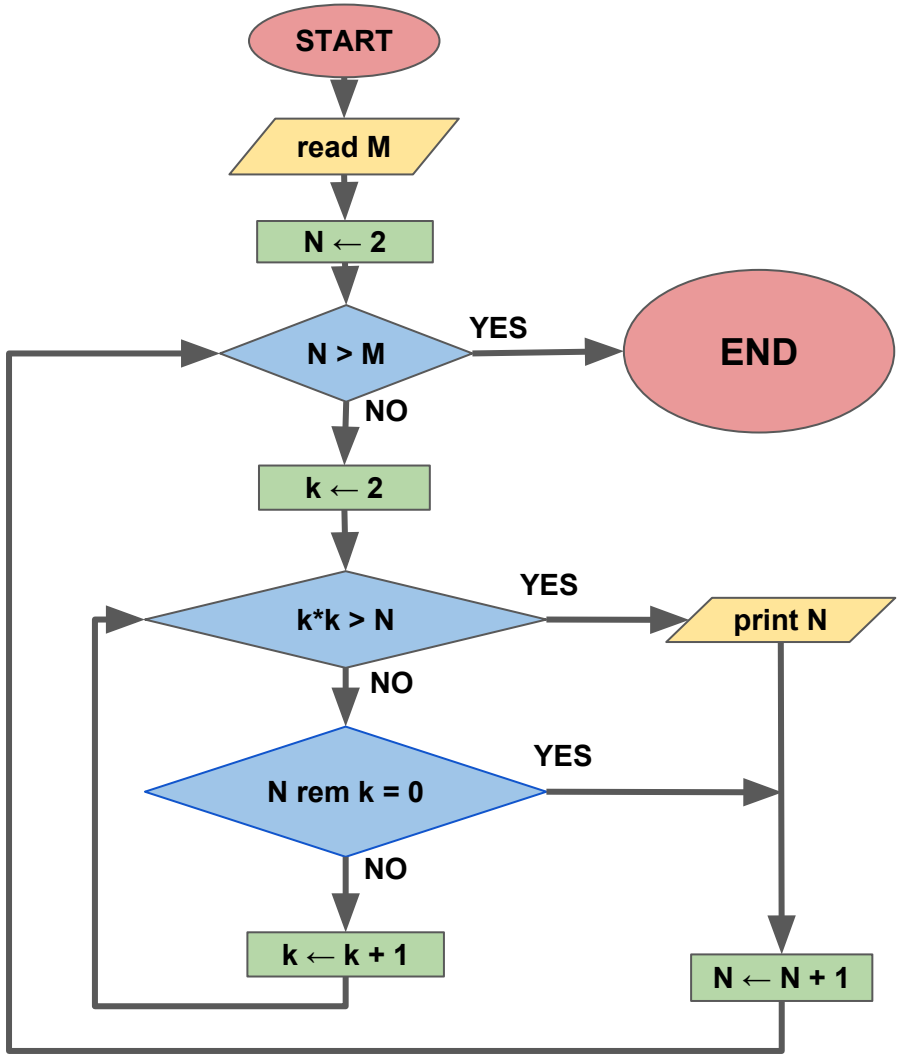
Be careful for autocorrection systems (homeworks, exam). Do not print prompts unless requested in the question.

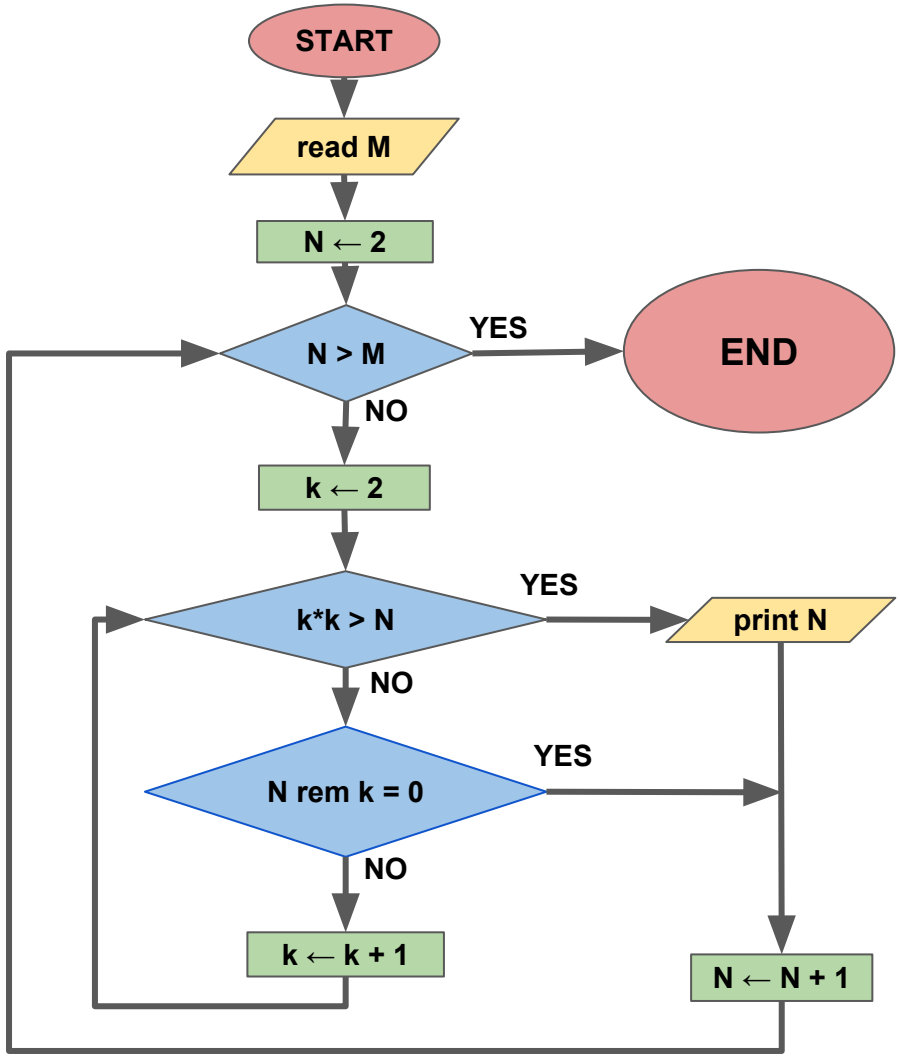
C keywords (reserved words)

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

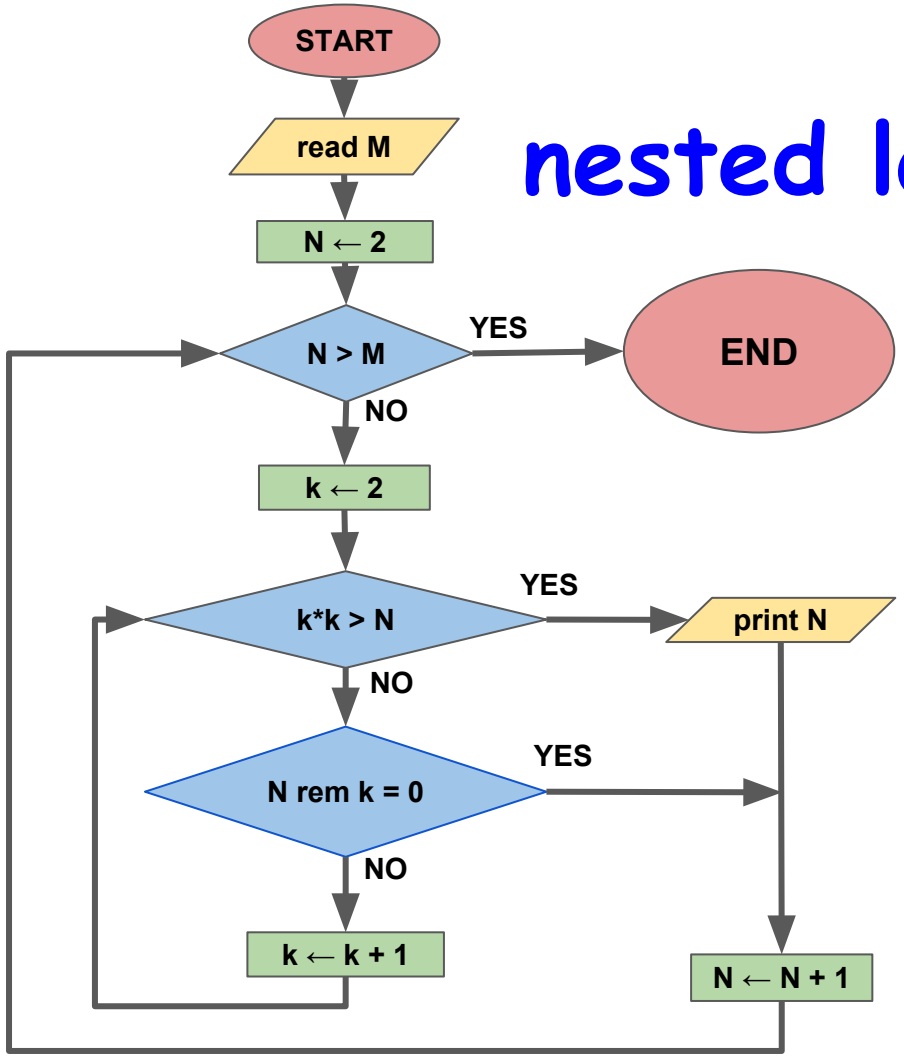
https://www.programiz.com/c-programming/list-all-keywords-c-language


START
read M
N ← 2
N > M
YES
END
NO
Print N if prime
N ← N + 1


START
read M
N ← 2
N > M
YES
END
NO
k ← 2
k*k > N
YES
print N
NO
N rem k = 0
YES
NO
k ← k + 1
N ← N + 1




```c
#include <stdio.h>

int main() {
  unsigned int N,M, 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;
  }
}
```



nested loop!

```c
#include <stdio.h>

int main() {
  unsigned int N,M, 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;
  }
}
```

Counter controlled iteration

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


   k = k + 1;
}
```

Counter-controlled iteration

adding up n numbers

Write a program reading students' scores in a class and printing their sum.

Counter-controlled iteration

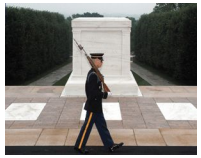
adding up n numbers

```
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);
```

Sentinel controlled iteration

```c
int main() {
  float a,sum;
  int n,k;

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

    if (a < 0)
      break;

    sum = sum + a;

    k++;
  }


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

  return 0;
}
```

Sentinel controlled loop



```c
int main() {
  float a,sum;
  int n,k;

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

    if (a < 0)
      break;

    sum = sum + a;

    k++;
  }


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

  return 0;
}
```

```c
int main() {
  float a,sum;
  int n,k;

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

    if (a < 0)
      break;

    sum = sum + a;

    k++;
  }


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

  return 0;
}
```

Average

```c
int main() {
  float a,sum;
  int n,k;

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

    if (a < 0)
      break;

    sum = sum + a;
    k++;
  }


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

  return 0;
}
```

Average

```c
int main() {
  float a,sum;
  int n,k;

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

    if (a < 0)
      break;

    sum = sum + a;
    k++;
  }


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

  return 0;
}
```

```c
int main() {
  float a,sum;
  int n,k;

  sum = 0;
  k = 0;

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

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


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

  return 0;
}
```