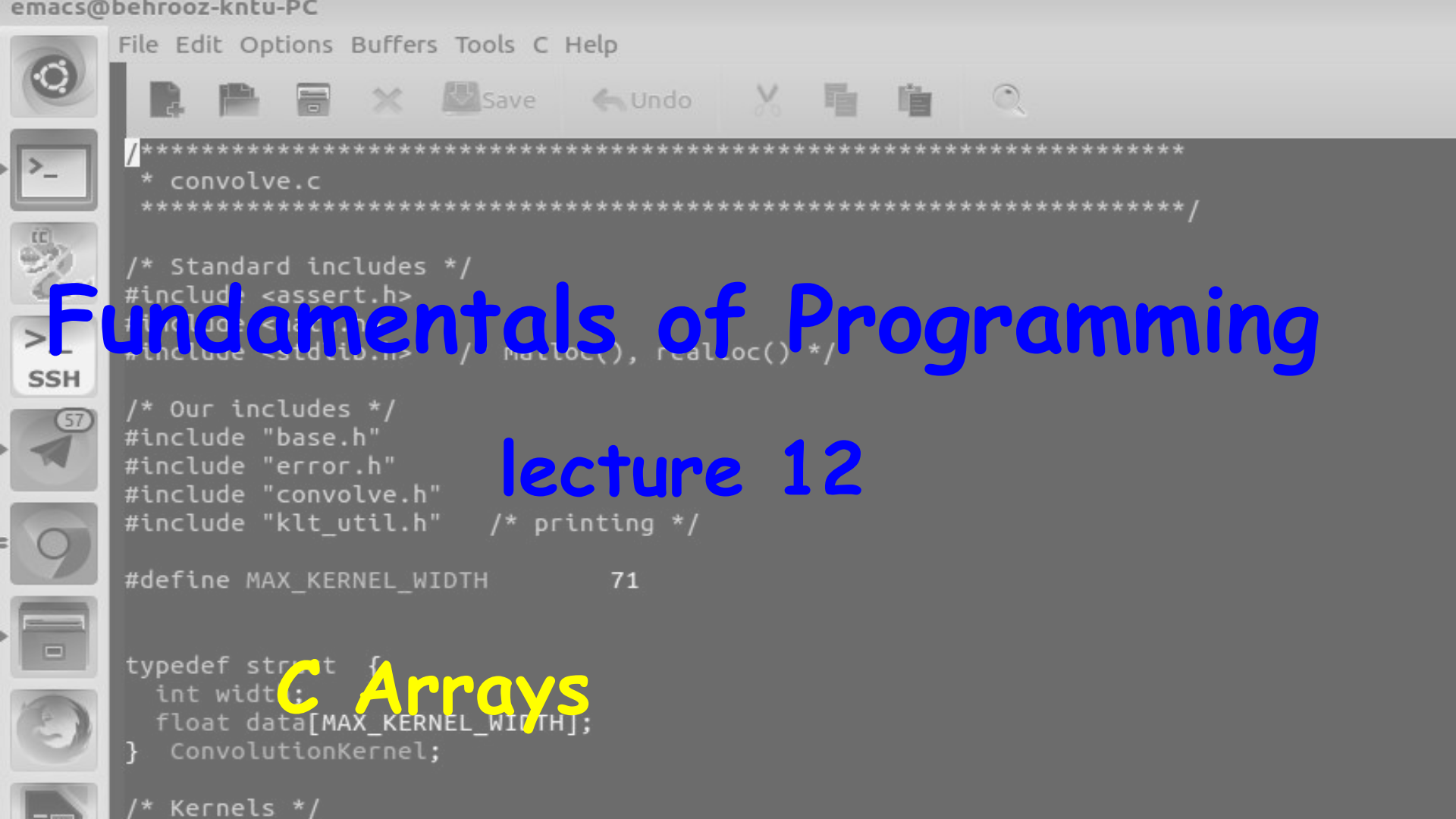

emacs@behrooz-kntu-PC
File Edit Options Buffers Tools C Help
Save
Undo
/* Standard includes */
/* Our includes */
#include "base.h"
#include "error.h"
#include "convolve.h"
#include "klt_util.h" /* printing */
#define MAX_KERNEL_WIDTH 71
typedef struct {
int width;
float data[MAX_KERNEL_WIDTH];
} ConvolutionKernel;
/* Kernels */
SSH
Fundamentals of Programming
lecture 12
C Arrays

Remember: count the ratings

prog11.c

```
r1=r2=r3=r4 = 0;
n = 0;
while (1) {
  printf("Enter rating: ");
  scanf("%d", &rating);

  if (rating == 1)
      r1++;
  else if (rating == 2)
      r2++;
  else if (rating == 3)
      r3++;
  else if (rating == 4)
      r4++;
  else if (rating == -1)
      break;
  else {
      puts("invalid number!");
      continue;
  }
```

prog11.c (cont.)

```
  n++;
}

printf("      Bad: %.1f%% \n",
        100*r1/(float)n);
printf("  Average: %.1f%% \n",
        100*r2/(float)n);
printf("     Good: %.1f%% \n",
        100*r3/(float)n);
printf("Excellent: %.1f%% \n",
        100*r4/(float)n);
```

1: Bad

2: Average

3: Good

4: Excellent

Remember: count the ratings

```c
int n, r,  r1, r2, r3, r4;

r1 = r2 = r3 = r4 = 0;
n = 0;
while (1) {
  printf("Enter rating: ");
  scanf("%d", &r);


  if (r== 1)
    r1++;
  else if (r== 2)
    r2++;
  else if (r== 3)
    r3++;
  else if (r== 4)
    r4++;
  else if (r== -1)
    break;
  else {
    puts("invalid number!");
    continue;
  }


  // write your code here
  n++;
}
```

1: Bad

2: Average

3: Good

4: Excellent

```c
finish = 0;
while (1) {
  printf("Enter rating: ");
  scanf("%d", &r);

  switch (r) {
  case 1:
    r1++;
    break;

  case 2:
    r2++;
    break;

  case 3:
    r3++;
    break;

  case 4:
    r4++;
    break;

  case -1:
    finish = 1;
    break;

  default:
    continue;
    break;

  }

  if (finish)
    break;

  n++;
}
```

What if there are 20 numbers?

```c
int n, r,  r1, r2, r3, r4;

r1 = r2 = r3 = r4 = 0;
n = 0;
while (1) {
  printf("Enter rating: ");
  scanf("%d", &r);


  if (r== 1)
    r1++;
  else if (r== 2)
    r2++;
  else if (r== 3)
    r3++;
  else if (r== 4)
    r4++;
  else if (r== -1)
    break;
  else {
    puts("invalid number!");
    continue;
  }


  // write your code here
  n++;
}
```

1:

2:

3:

:

20:

```c
finish = 0;
while (1) {
  printf("Enter rating: ");
  scanf("%d", &r);

  switch (r) {
  case 1:
    r1++;
    break;

  case 2:
    r2++;
    break;

  case 3:
    r3++;
    break;

  case 4:
    r4++;
    break;

  case -1:
    finish = 1;
    break;

  default:
    continue;
    break;

  }

  if (finish)
    break;

  n++;
}
```

What if there are 100 numbers?

```
int n, r,  r1, r2, r3, r4;

r1 = r2 = r3 = r4 = 0;
n = 0;
while (1) {
  printf("Enter rating: ");
  scanf("%d", &r);


  if (r== 1)
    r1++;
  else if (r== 2)
    r2++;
  else if (r== 3)
    r3++;
  else if (r== 4)
    r4++;
  else if (r== -1)
    break;
  else {
    puts("invalid number!");
    continue;
  }


  // write your code here
  n++;
}
```

1:

2:

3:

:

100:

```
finish = 0;
while (1) {
  printf("Enter rating: ");
  scanf("%d", &r);

  switch (r) {
  case 1:
    r1++;
    break;

  case 2:
    r2++;
    break;

  case 3:
    r3++;
    break;

  case 4:
    r4++;
    break;

  case -1:
    finish = 1;
    break;

  default:
    continue;
    break;

  }

  if (finish)
    break;

  n++;
}
```

What if there are a million numbers?

```
int n, r,  r1, r2, r3, r4;

r1 = r2 = r3 = r4 = 0;
n = 0;
while (1) {
  printf("Enter rating: ");
  scanf("%d", &r);


  if (r== 1)
    r1++;
  else if (r== 2)
    r2++;
  else if (r== 3)
    r3++;
  else if (r== 4)
    r4++;
  else if (r== -1)
    break;
  else {
    puts("invalid number!");
    continue;
  }


  // write your code here
  n++;
}
```

1:

2:

3:

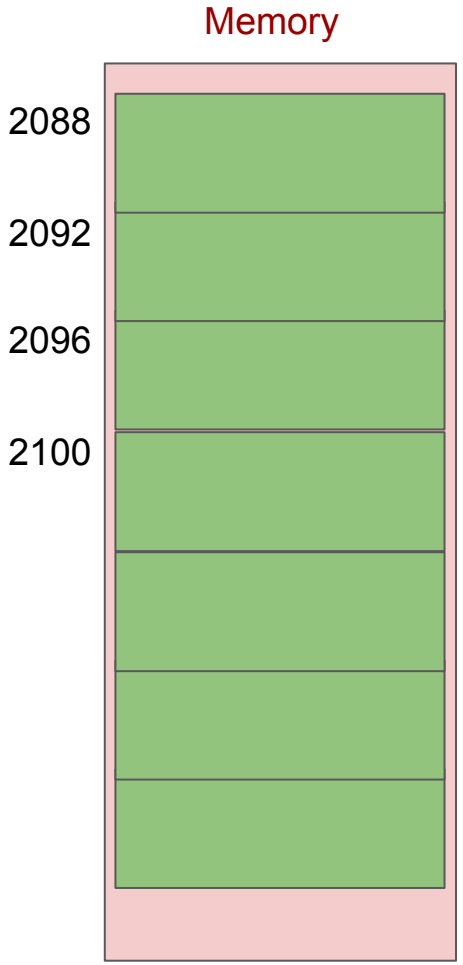
:

1000000:

```
finish = 0;
while (1) {
  printf("Enter rating: ");
  scanf("%d", &r);

  switch (r) {
  case 1:
    r1++;
    break;

  case 2:
    r2++;
    break;

  case 3:
    r3++;
    break;

  case 4:
    r4++;
    break;

  case -1:
    finish = 1;
    break;

  default:
    continue;
    break;

  }

  if (finish)
    break;

  n++;
}
```

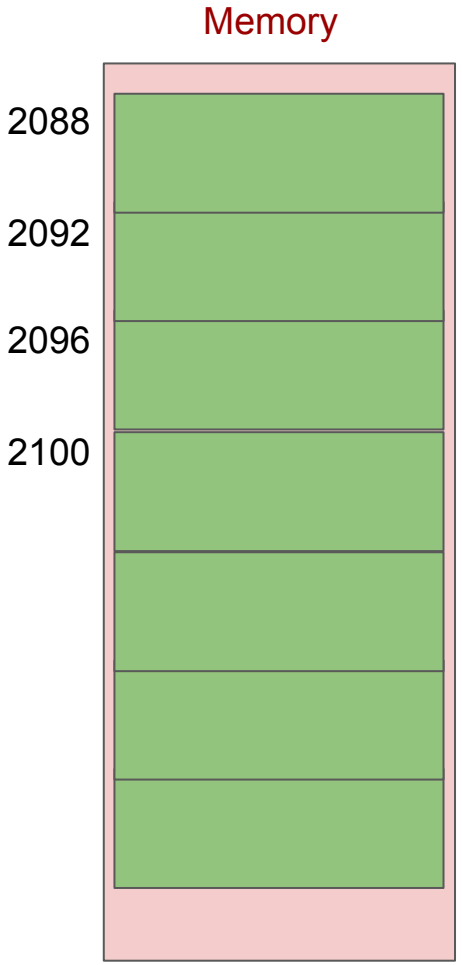
C Arrays

```
int a[7];
```



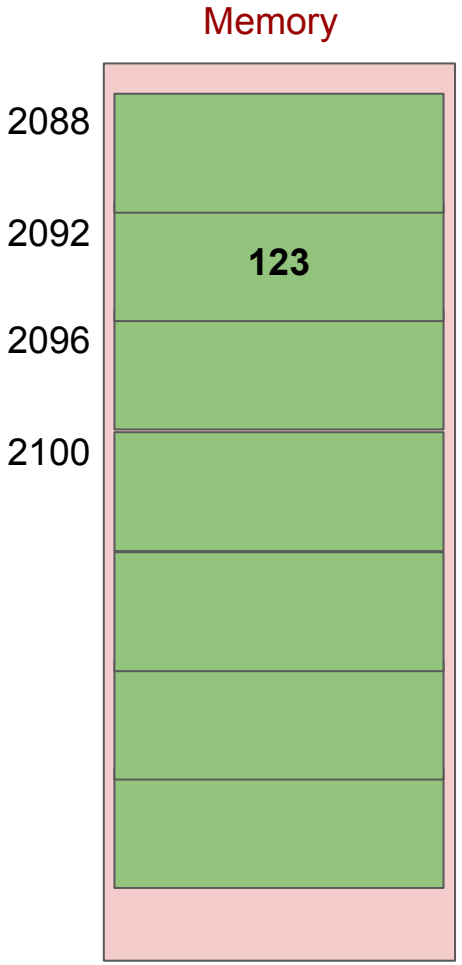
C Arrays

```
int a[7];

a[1] = 123;
```



C Arrays

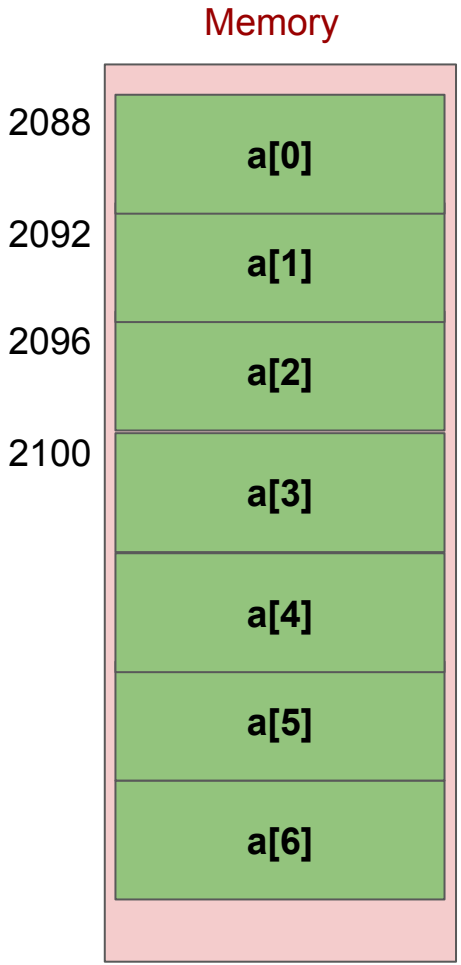
```
int a[7];

a[1] = 123;
```



C Arrays

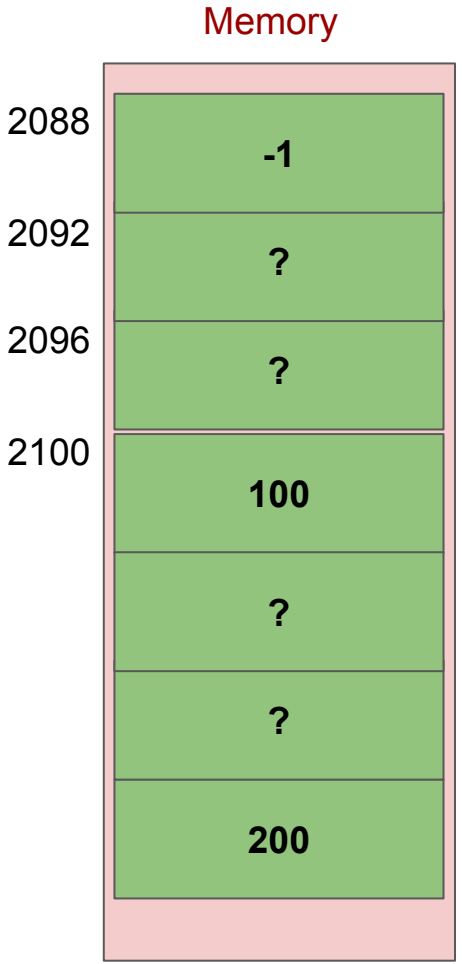
```
int a[7];

a[1] = 123;
```

index,
subscript



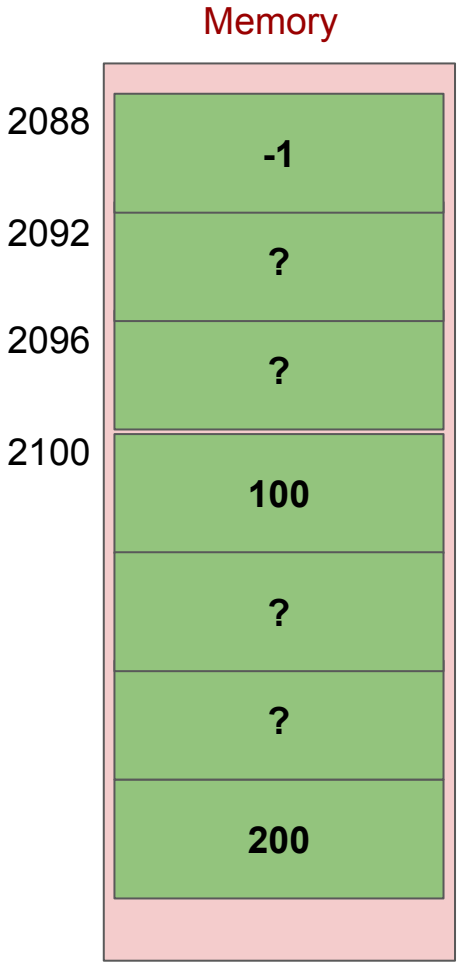
C Arrays

```
int a[7];

a[0] = -1;
a[3] = 100;
a[6] = 200;
```



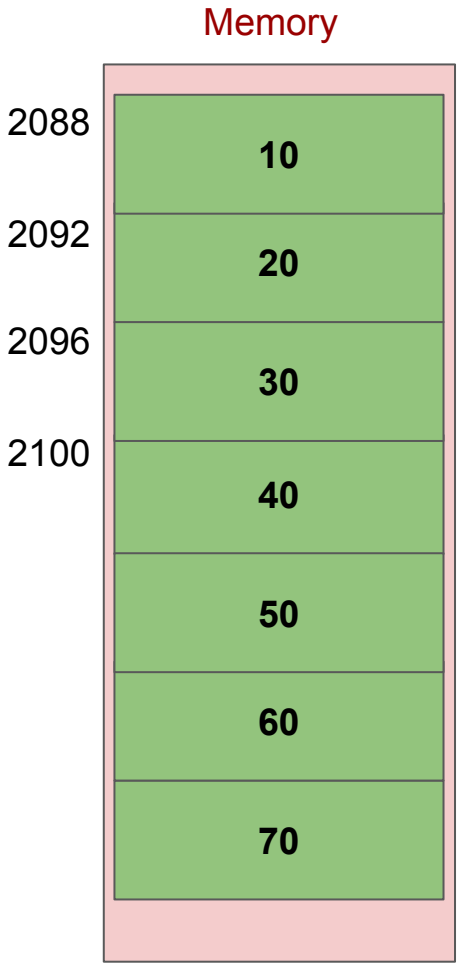
C Arrays

```
int a[7];

a[0] = -1;
a[3] = 100;
a[6] = 200;

a[7] = 700;
```



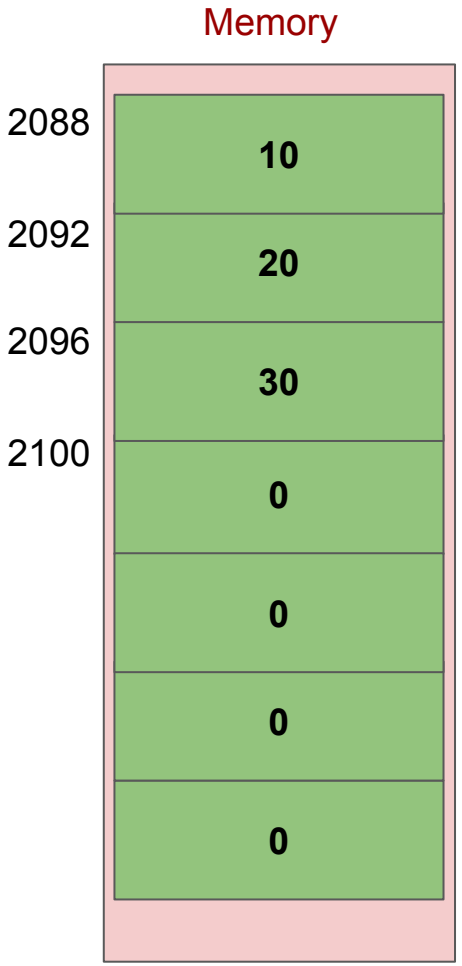
Initialize Arrays

```
int a[7] = {10,20,30,40,50,60,70};
```



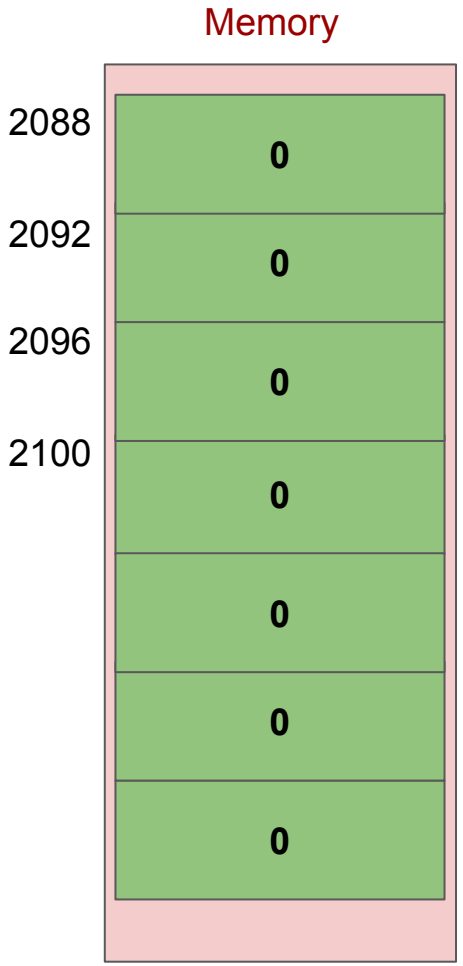
Initialize Arrays

```
int a[7] = {10,20,30};
```



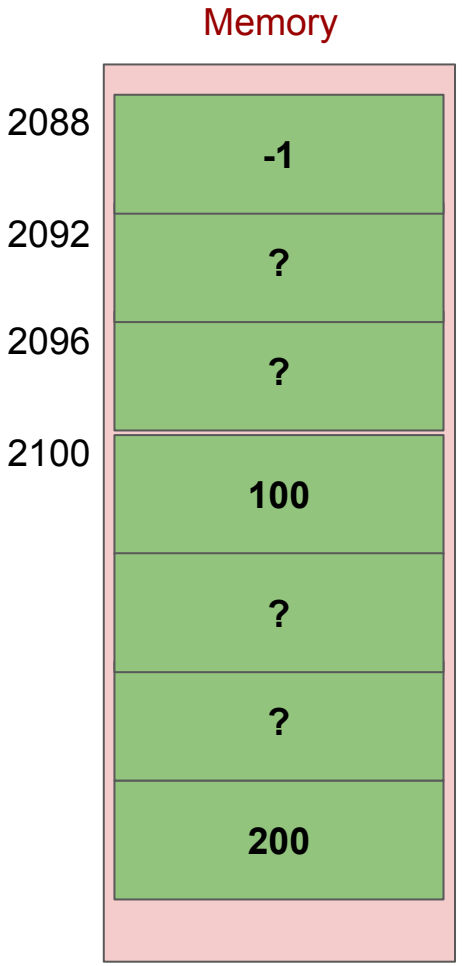
Initialize Arrays

```
int a[7] = {0};
```



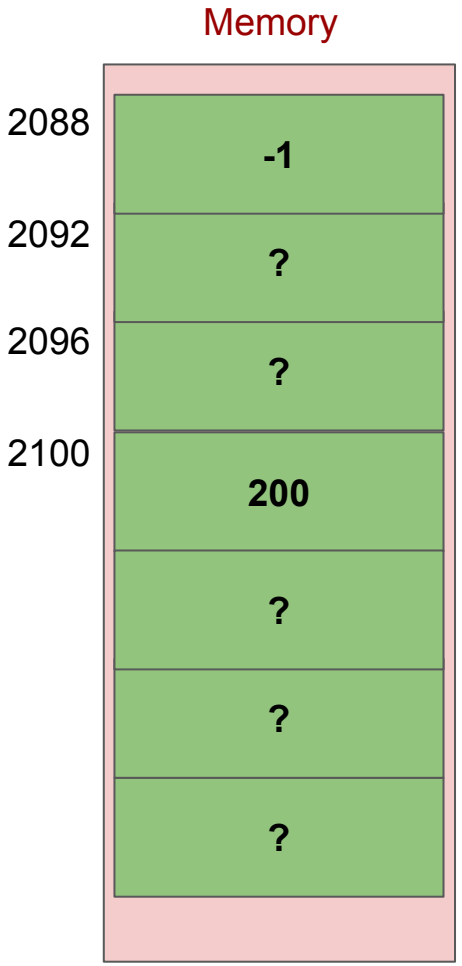
So what?

```
int a[7];

a[0] = -1;
a[3] = 100;
a[6] = 200;
```



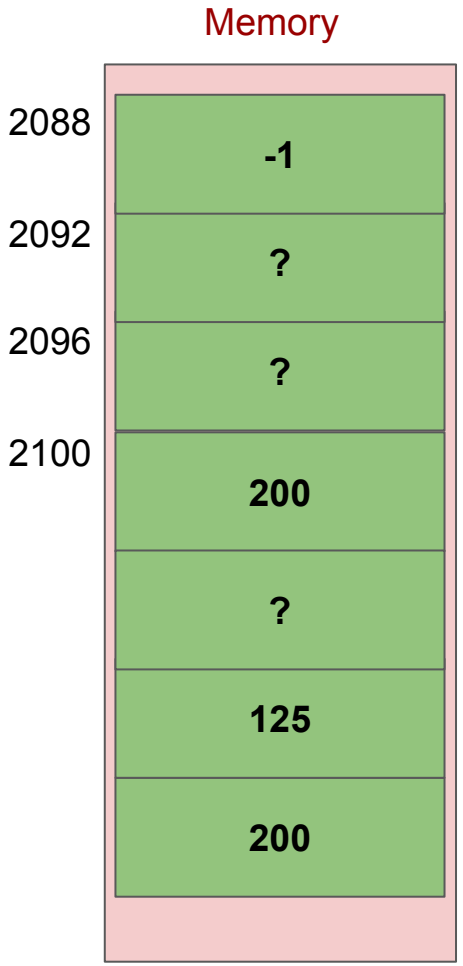
difference with ordinary variables

```
int a[7];

a[0] = -1;
a[3] = 100;
a[6] = 200;

int i = 3;
a[i] = 200;
```



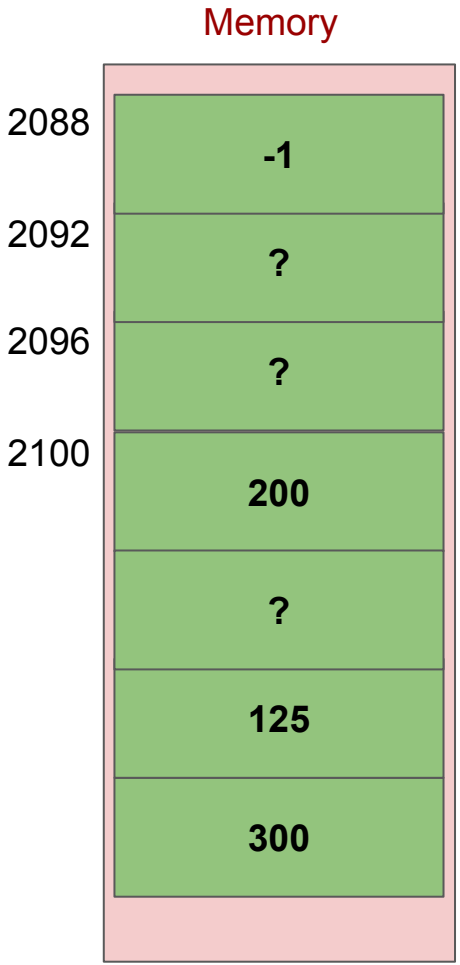
difference with ordinary variables

```
int a[7];

a[0] = -1;
a[3] = 100;
a[6] = 200;

int i = 3;
a[i] = 200;

i += 2;
a[i] = i*i*i;
```



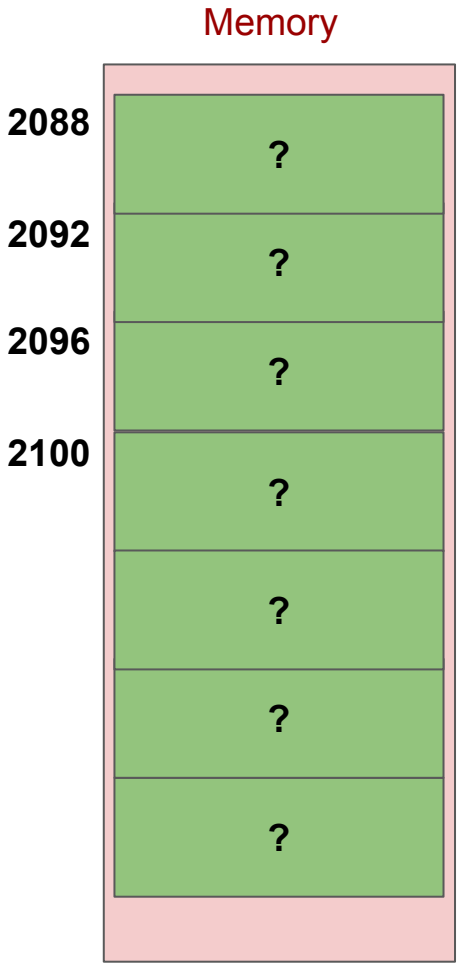
difference with ordinary variables

```
int a[7];

a[0] = -1;
a[3] = 100;
a[6] = 200;

int i = 3;
a[i] = 200;

i += 2;
a[i] = i*i*i;

a[i+1] = 300;
```



difference with ordinary variables

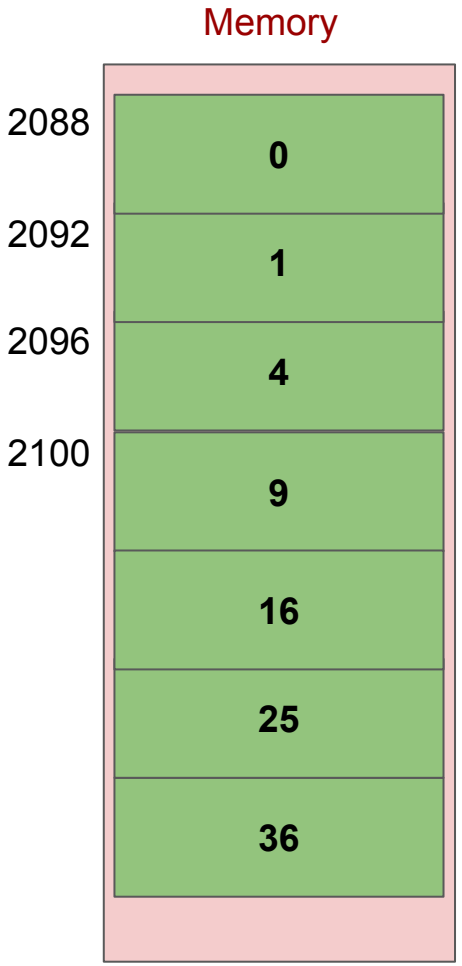
```
int a[7];

for (int i = 0; i < 7; i++) {
    a[i] = i*i;
}
```



difference with ordinary variables

```
int a[7];

for (int i = 0; i < 7; i++) {
    a[i] = i*i;
}
```

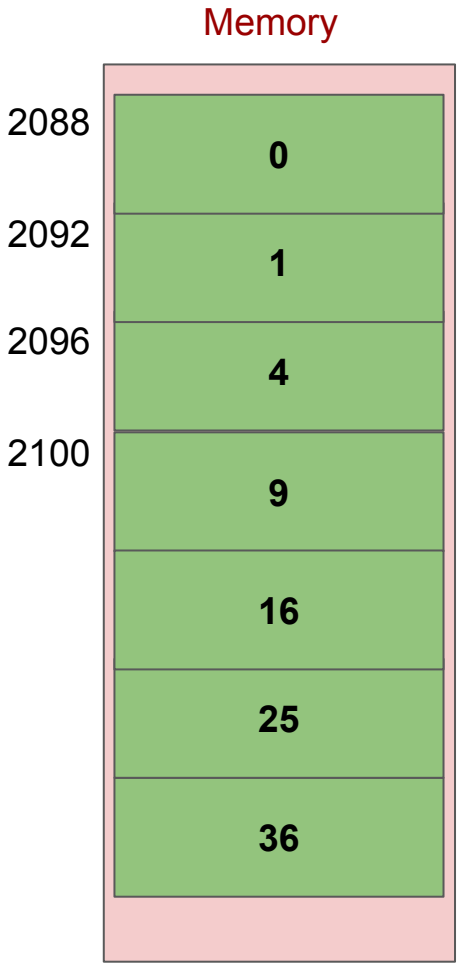
array1.c



difference with ordinary variables

```
int a[7];

for (int i = 0; i < 7; i++) {
    a[i] = i*i;
}

for (int i = 0; i < 7; i++) {
    printf("%d\n", a[i]);
}
```

array1.c



Back to our problem

get students scores (all integers {0,1,2,...,19,20}, no fractions). For each number 0,1,2,....,20 print the number of students with this score.

```
while (1) {
  int g;

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

  if (g < 0 || g > 20)
    break;



}
```

Back to our problem

get students scores (all integers {0,1,2,...,19,20}, no fractions). For each number 0,1,2,....,20 print the number of students with this score.

```
int count[20];

while (1) {
  int g;

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

  if (g < 0 || g > 20)
    break;



}
```

Back to our problem

get students scores (all integers {0,1,2,...,19,20}, no fractions). For each number 0,1,2,....,20 print the number of students with this score.

```
int count[21];

while (1) {
  int g;

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

  if (g < 0 || g > 20)
    break;



}
```

Back to our problem

get students scores (all integers {0,1,2,...,19,20}, no fractions). For each number 0,1,2,....,20 print the number of students with this score.

```
int count[21];

for (int i = 0; i < 21; i++)
  count[i] = 0;


while (1) {
  int g;

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

  if (g < 0 || g > 20)
    break;



}
```

Back to our problem

get students scores (all integers {0,1,2,...,19,20}, no fractions). For each number 0,1,2,....,20 print the number of students with this score.

```
int count[21];

for (int i = 0; i < 21; i++)
  count[i] = 0;


while (1) {
  int g;

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

  if (g < 0 || g > 20)
    break;

  count[g] = count[g] + 1;

}
```

Back to our problem

get students scores (all integers {0,1,2,...,19,20}, no fractions). For each number 0,1,2,....,20 print the number of students with this score.

```c
int count[21];

for (int i = 0; i < 21; i++)
  count[i] = 0;


while (1) {
  int g;

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

  if (g < 0 || g > 20)
    break;

  count[g]++;

}

for (int i = 0; i < 21; i++)
  printf("%2d: %d\n", i, count[i]);
```

countarray3.c

```c
#include <stdio.h>



int main() {

  int count[21];

  for (int i = 0; i < 21; i++)
    count[i] = 0;


  while (1) {
    int g;

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

    if (g < 0 || g >= 21)
      break;

    count[g]++;

  }

  for (int i = 0; i < 21; i++)
    printf("%2d: %d\n", i, count[i]);

  return 0;
}
```

Change the code so that grades are between 0 and 100.

```c
#include <stdio.h>


int main() {

  int count[21];

  for (int i = 0; i < 21; i++)
    count[i] = 0;


  while (1) {
    int g;

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

    if (g < 0 || g >= 21)
      break;

    count[g]++;

  }

  for (int i = 0; i < 21; i++)
    printf("%2d: %d\n", i, count[i]);

  return 0;
}
```

Change the code so that grades are between 0 and 100.

```c
#include <stdio.h>


int main() {

  int count[21];

  for (int i = 0; i < 21; i++)
    count[i] = 0;


  while (1) {
    int g;

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

    if (g < 0 || g >= 21)
      break;

    count[g]++;

  }

  for (int i = 0; i < 21; i++)
    printf("%2d: %d\n", i, count[i]);

  return 0;
}
```

```c
#include <stdio.h>

#define N 21

int main() {

  int count[N];

  for (int i = 0; i < N; i++)
    count[i] = 0;


  while (1) {
    int g;

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

    if (g < 0 || g >= N)
      break;

    count[g]++;

  }

  for (int i = 0; i < N; i++)
    printf("%2d: %d\n", i, count[i]);

  return 0;
}
```

countarray4.c

```c
#include <stdio.h>

#define N 21

int main() {

  int count[N];

  for (int i = 0; i < N; i++)
    count[i] = 0;


  while (1) {
    int g;

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

    if (g < 0 || g >= N)
      break;

    count[g]++;

  }

  for (int i = 0; i < N; i++)
    printf("%2d: %d\n", i, count[i]);

  return 0;
}
```

countarray4.c

Is N a variable?

```c
#include <stdio.h>

#define N 21

int main() {

  int count[N];

  for (int i = 0; i < N; i++)
    count[i] = 0;


  while (1) {
    int g;

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

    if (g < 0 || g >= N)
      break;

    count[g]++;

  }

  for (int i = 0; i < N; i++)
    printf("%2d: %d\n", i, count[i]);

  return 0;
}
```

countarray4.c

Is N a variable?

Let's look at the preprocessor output:

$ gcc -E countarray4.c

```
int count[21];

for (int i = 0; i < N; i++)
  count[i] = 0;
```

```
int count[21];

for (int i = 0; i < N; i++)
  count[i] = 0;
```

```
int count[21] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
```

```
int count[21];

for (int i = 0; i < N; i++)
  count[i] = 0;
```

```
int count[21] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
```

```
int count[21] = {0};
```

```
int count[21];

for (int i = 0; i < N; i++)
  count[i] = 0;
```

```
int count[21] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
```

```
int count[21] = {0};
```

```
#define N 21

int main() {

  int count[N] = {0};
```

```c
int count[21];

for (int i = 0; i < N; i++)
  count[i] = 0;
```

```c
int count[21] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
```

```c
int count[21] = {0};
```

```c
#define N 21

int main() {

  int count[N] = {0};
```

Only works for initializing to zero!

```c
#include <stdio.h>

#define N 21

int main() {

  int count[N];

  for (int i = 0; i < N; i++)
    count[i] = 0;


  while (1) {
    int g;

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

    if (g < 0 || g >= N)
      break;

    count[g]++;

  }

  for (int i = 0; i < N; i++)
    printf("%2d: %d\n", i, count[i]);

  return 0;
}
```

countarray4.c

Draw a Histogram instead.

```
 0: ****
 1: *
 2: *****
 3:
 4:
 5:
 6:
 7:
 8: **
 9: *
10:
```

```c
#include <stdio.h>

#define N 21

int main() {

  int count[N];

  for (int i = 0; i < N; i++)
    count[i] = 0;


  while (1) {
    int g;

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

    if (g < 0 || g >= N)
      break;

    count[g]++;

  }

  for (int i = 0; i < N; i++)
    printf("%2d: %d\n", i, count[i]);

  return 0;
}
```

countarray4.c

```c
#include <stdio.h>

#define N 21

int main() {

  int count[N] = {0};

  while (1) {
    int g;

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

    if (g < 0 || g >= N)
      break;

    count[g]++;

  }

  for (int i = 0; i < N; i++) {
    printf("%2d: ", i);

    for (int j = 0; j < count[i]; j++)
      printf("*");

    printf("\n");
  }

  return 0;
}
```

countarrayhist.c

```
 0: ****
 1: *
 2: *****
 3:
 4:
 5:
 6:
 7:
 8: **
 9: *
10:
```

```c
#include <stdio.h>

#define N 21

int main() {

  int count[N] = {0};

  while (1) {
    int g;

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

    if (g < 0 || g >= N)
      break;

    count[g]++;

  }

  for (int i = 0; i < N; i++) {
    printf("%2d: ", i);

    for (int j = 0; j < count[i]; j++)
      printf("*");

    printf("\n");
  }

  return 0;
}
```

countarrayhist.c

Roll a dice N times!

```c
#include <stdio.h>
#include <stdlib.h>
#define SIZE 6

int main() {
  int n;
  int count[SIZE+1];

  for (int i = 1; i <= SIZE; i++)
    count[i] = 0;

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

  for (int i = 0; i < n; i++) {
    int g = rand()%SIZE + 1;
    count[g]++;
  }

  for (int i = 1; i <= SIZE; i++)
    printf("%2d: %.1f%%\n", i, 100*count[i]/ (double)n);

  return 0;
}
```

Exercise

Write a program that prints the sum of the elements of an array.

```c
#include <stdio.h>

#define N 20

int main() {

  int a[N] = {1,3,4,5,6,1,2,3,4,5,6,7,8,1,3,1,4,4,19,20};
  int sum = 0;



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

  return 0;
}
```

Exercise

Write a program that prints the sum of the elements of an array.

```c
#include <stdio.h>

#define N 20

int main() {

  int a[N] = {1,3,4,5,6,1,2,3,4,5,6,7,8,1,3,1,4,4,19,20};
  int sum = 0;

  for (int i = 0; i < N; i++)
    sum += a[i];


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

  return 0;
}
```

sumarray.c

More on array definition

```
int a[12], b[13];
```

More on array definition

```
int a[12], b[13];
```

is the same as:

```
int a[12];
int b[13];
```

More on array definition

```
int a,b[13];
```

More on array definition

```
int a,b[12];
```

is the same as:

```
int a;
int b[12];
```

More on array definition

```
int a[] = {1,3,6,10};
```

More on array definition

```
int a[] = {1,3,6,10};
```

is the same as:

```
int a[4] = {1,3,6,10};
```