Lecture 10
Dynamic memory allocation for 2D arrays

and examples

Dynamic memory allocation

recall

```
#include <stdlib.h>

void* malloc (unsigned int size);
```

1. Allocates a block of **size** bytes of memory
2. Returns a pointer to the beginning of that block
3. The content allocated block of memory is not initialized
4. *size t* is *unsigned int*
5. For each malloc there needs to be a single *free*

```
type * p = (type*)malloc(size);
free(p)
```

6. **After** we are done with using the memory

Dynamic memory allocation

example

Read the size from keyboard and allocate memory to store doubles.

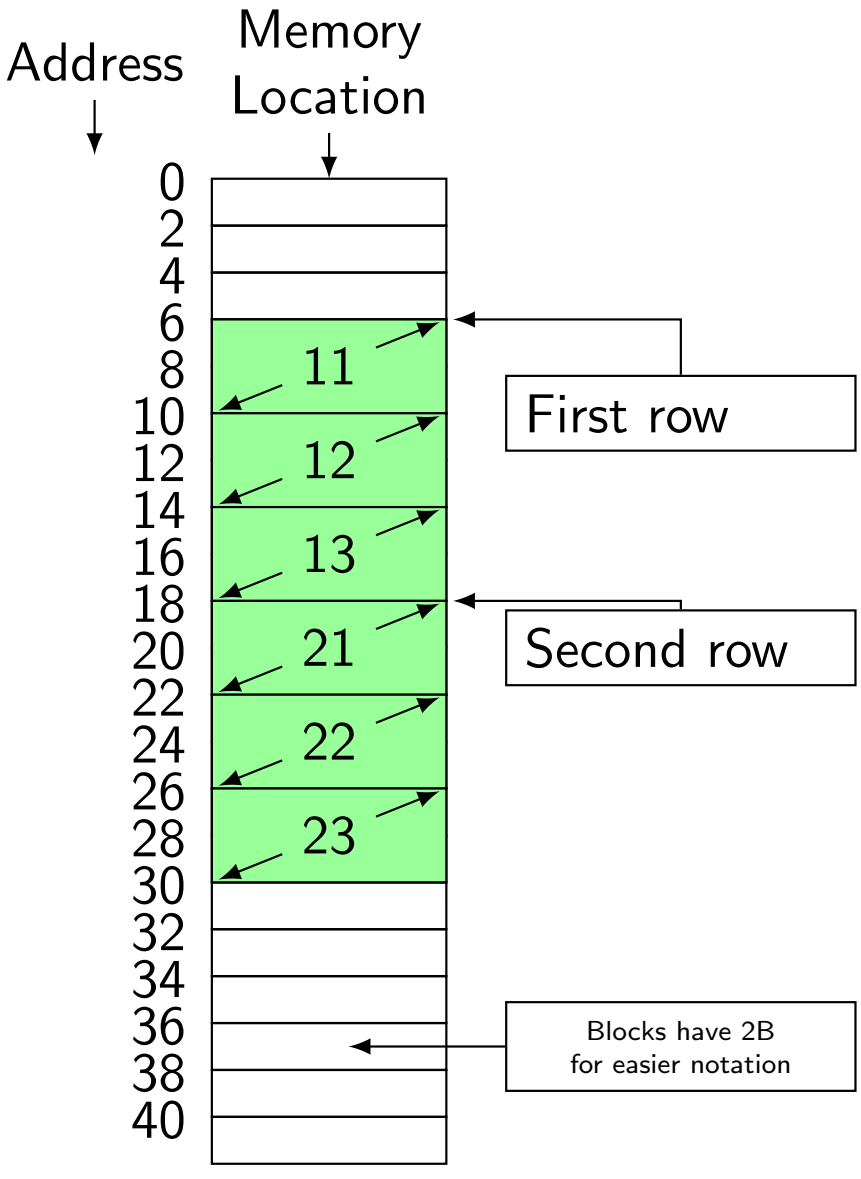
```
#include <stdio.h>
#include <stdlib.h>

int mian(){
        int n;
        scanf("%d", &n);
        double *w = (double*)malloc(n*sizeof(double));
        free(w);
}
```

This is a 1D array, how about 2D?

2D arrays

Recall the static 2D



- Indexing is from 0 to size-1
- Storage is row based
- Array is stored row after row

Example:

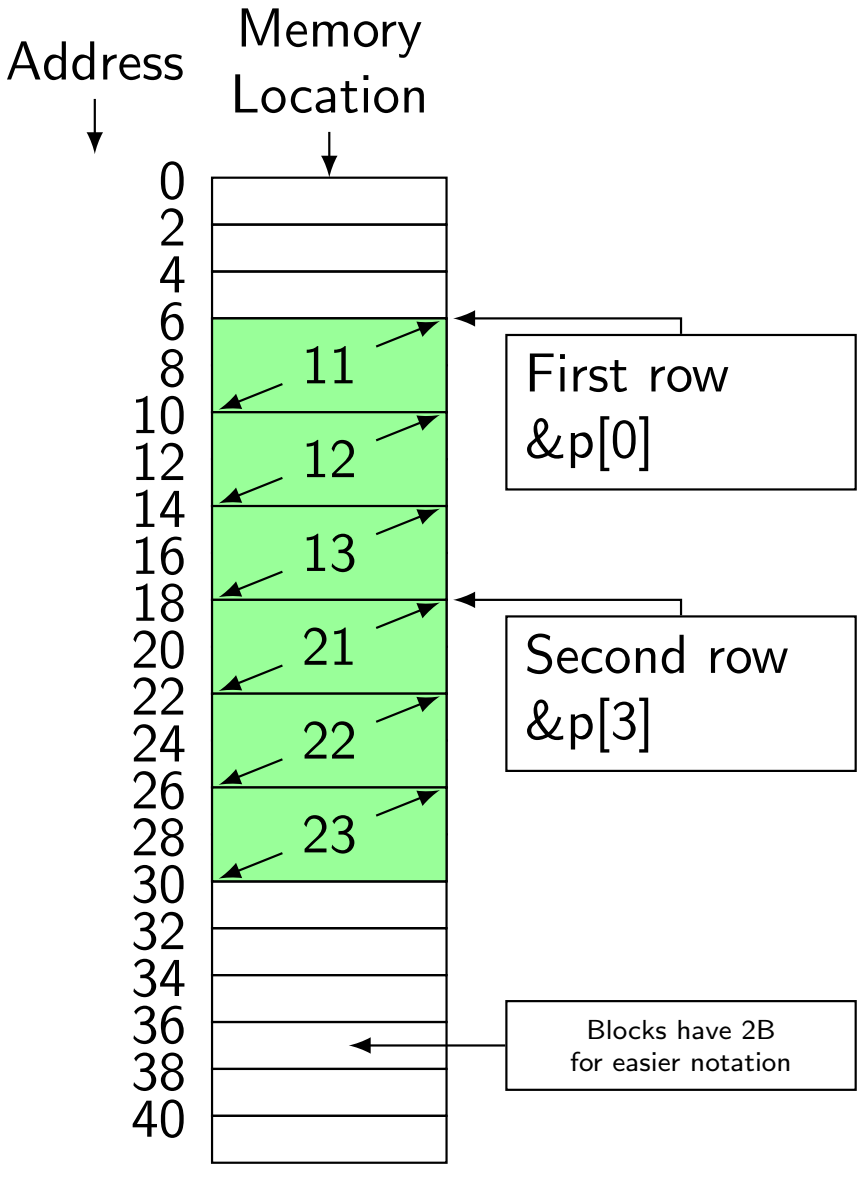
```
int tab[2][3];

tab[0][0]=11;
tab[0][1]=12;
tab[0][2]=13;

tab[1][0]=21;
tab[1][1]=22;
tab[1][2]=23;
```

11	12	13
21	22	23

2D arrays




```
int *p=(int*)malloc(24);

p[0] = 11;
p[1] = 12;
p[2] = 13;
p[3] = 21;
p[4] = 22;
p[5] = 23;

free(p);
```

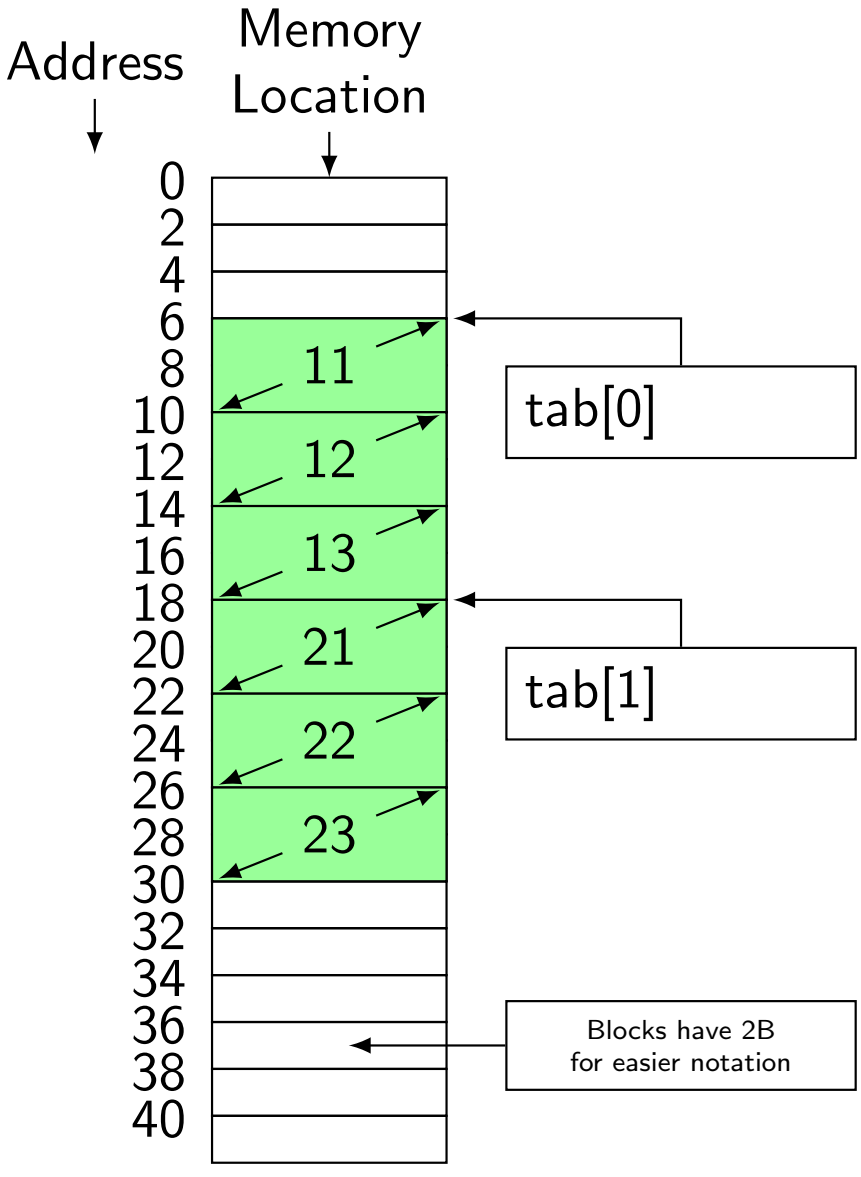
2D arrays
Pointers to pointers

```
int a = 5;
int *p=&a;
int **pp=&p;// !!!

printf("%d", a);//simple
printf("%d", *p);//still simple
printf("%d", *(*pp));//?

...
printf("%d", a);//simple
printf("%d", p[0]);//still simple
printf("%d", pp[0][0]));//?
```

2D arrays




```
int *tab[2];//what is tab?
int *p=(int*)malloc(24);

tab[0] = &p[0];//but also tab[0]=p;
p[0] = 11;
p[1] = 12;
p[2] = 13;
tab[1] = &p[3];//but also tab[1]=p+3;
p[3] = 21;
p[4] = 22;
p[5] = 23;

tab[0][0] = ?

free(p);
```

2D arrays
Pointers to pointers

```
int **A = (int**)malloc(number_of_rows * szieof(int*));
```

1.
```
for(int i=0; i<number_of_rows; ++i)
        A[i] = (int*)malloc(number_of_columns * sizeof(int));
```

2.
```
int *p =(int*)malloc(collumns*rows*sizeof(int));
for(int i=0; i<number_of_rows; ++i)
        A[i] = p+i*number_of_collumns;
```

1.
```
for(int i=0; i<number_of_rows; ++i)
        free(A[i]);
free(A);
```

2.
```
free(p);
free(A);
```

Example...