# Computer Programming 1 Lab

2022-11-17

# Outline

- Pointer

- recursive function

- Exercise 7

# Pointer

# Pointer

- Array

code

```c
int arr[10];

printf("%d", arr[5]);
printf("%d", *(arr+5));
```

# Pointer

- 2D Array

```c
int arr[10][10];

printf("%d", arr[2][3]);
printf("%d", *(*(arr+2)+3));
```

# Pointer

**malloc**

# Pointer

## malloc

- 標頭檔 #include <stdlib.h>

- 使用方法

  資料型態 *Ptr;*
  *Ptr = (資料型態*)malloc(sizeof(資料型態) * 數量)

# Pointer

## malloc

```c
int arr[10];
int *arrPtr = (int*)malloc(sizeof(int) * 10);

printf("%d", arr[5]);
printf("%d", *(arrPtr+5));
```

# Pointer

## malloc

```c
int *arrPtr = (int*)malloc(sizeof(int) * 10);
bool *arrPtr1 = (bool*)malloc(sizeof(bool) * 10);
short *arrPtr2 = (short*)malloc(sizeof(short) * 10);
float *arrPtr3 = (float*)malloc(sizeof(float) * 10);
double *arrPtr4 = (double*)malloc(sizeof(double) * 10);
long long *arrPtr5 = (long long*)malloc(sizeof(long long) * 10);
unsigned long long *arrPtr6 = (unsigned long long*)malloc(sizeof(unsigned long long) * 10);
```

# Pointer

## malloc

```c
int *arrPtr = malloc(sizeof(int) * 10);

for(int i = 0; i < 10; i++){
    arrPtr[i] = i;
}

//free after use
free(arrPtr);
```

# Pointer

## malloc

```c
//int arr[2][3];

int **arr = (int**)malloc(sizeof(int*) * 2);
for (int i = 0; i < 2; i++)
    *(arr+i) = (int*)malloc(sizeof(int) * 3);

printf("%d", *(*(arr+1)+2));
```

# Pointer

## malloc

```c
//int arr[2][3];

int **arr = (int**)malloc(sizeof(int*) * 2);
for (int i = 0; i < 2; i++)
    *(arr+i) = (int*)malloc(sizeof(int) * 3);

//free after use
for(int i = 0; i < 2; i++) {
    free(*(arr+i));
}
free(arr);
```

# Sort Function

# qsort

void qsort(void* base, size_t n, size_t size, int (*cmp)(const void*, const void*))

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

int mycomp(const void* a, const void* b){
    int A = *(int*)a;
    int B = *(int*)b;
    return A - B;
}

int main(){
    int arr[10] = {0, 9, 8, 7, 2, 5, 4, 1, 3, 6};
    qsort(arr, 10, sizeof(int), mycomp);
    for(int i = 0; i < 10; i++){
        printf("%d ", arr[i]);
    }
    printf("\n");
    return 0;
}
```

# Exercise 7

# Any Questions?