C5 Array

5.1 Declaring Array

```
int tests[5];
```

```
const int SIZE = 5;
int tests[SIZE];
```

5.2 Store with string

```
char fname[6] = "Henry";
```

H	e	n	r	y	\0	*Total*
0	1	2	3	4	5	6 space

5.3 Different btw test[i++] and test[i]++

```
int test[]={1,2,5};

test[2]++;  // =2+1=3
test[2++];  // =test[3]=5
```

5.4 Finding Highest and Low *(Comparing)*

```
///Highest
int highest = number[0];
for (int cnt=1; cnt<SIZE; cnt++)
{
    if (number[cnt] > highest)
    {
        highest = number[cnt];
    }
}

///Lowest
int lowest = number[0];
for (int cnt=1; cnt<SIZE; cnt++)
{
    if (number[cnt] < lowest)
    {
        lowest = number[cnt];
    }
}
```

5.5 Parallal Array

```
int test1[2];
int test2[2];

for(.....)
    int total[i] = test1[i]*test2[i];
```

5.6 Array as Function Argument

function call

```
showScores(test);
```

Prototype and Header

```
void showScores (int []);          //Prototype
void showScores (int test[]);      //Header
```

In the Function

```
int main()
{
int test[]= {1,2,3,4}
changeValue(test);
cout test[3];           // Here will come out 9,not 4 as change in the void function
}

void changeValue(int a[])
{
a[3] = 9;
}
```

Data Type

if array is *double value*, use *double dt*

if array is *int value*, use *int dt*

5.7 Two Dimension Array

Declare

```
// main function
const int ROWS = 4, COLS = 3;
int a [ROWS][COLS];

//Prototype
void getExam(int[][],int);

//Header
void getExam(int[][COLS], int rows)

//Function call
getExam(a,4);

// In the first [] of prototype and Header must be empty, then for the row declare will be at th
```

Representation

Row\Column	0	1	2
0	a[0][0]	a[0][1]	a[0][2]
1	a[1][0]	a[1][1]	a[1][2]
2	a[2][0]	a[2][1]	a[2][2]
3	a[3][0]	a[3][1]	a[3][2]

5.8 Array with String

```
const int NAMES = 3; SIZE = 10;
char student[NAMES][SIZE] = {"Ann", "Bill", "Cindy"}    // SIZE means the size string can hold.

cout << student[2]; // Bill
```

Grammar Mistake

```
test[0] = 79;
test[1] = 80;
test[3] = test[0] + test[1];  // test[3] become 159

cout << test;       //not legal
cout << test[3];    //legal (must be individual)

-----------------------------------------------------------

char fName[] = "Henry";
cout << fName << endl; //only char can direct cout array


int test[]= {1,2,3};        //legal
int test[3];                //legal
int test[3] = {1,2,3}       //legal

//must have either size declare,[2] or [];
```