

Summary of Lesson 6

Array

- . Array is series of datas of same kind
- . declaration of array is always with array bracket (e.g. int myArray[20])
- . in C language, a string is an array of char
- . index number starts with **0** not **1**
- . 2D, 3D ...nD array are also possible

lesson6_1.c

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

```
int main(void)
{
    int i;
    int array[5] = {1,2,4,8,16};
    printf("second element of array:%d\n",array[1]); // refer one element
    for(i = 0;i<5;i++) // print all element
    {
        printf("%d : %d\n",i,array[i]);
    }
    array[1] = 99; // change an element
    printf("print changed array:\n");
    for(i = 0;i<5;i++) // print again
    {
        printf("%d : %d\n",i,array[i]);
    }
    return 0;
}
```

lesson6_2.c

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

```
int main(void)
{
    int i = 0;
    char myName[] = "chikashi";
    int gap;
    printf("name:");
    for(i = 0;i<8;i++)
    {
        printf("%c",myName[i]);
    }
}
```

```

printf("\nreversed:");
for(i = 7;i>=0;i--)
{
    printf("%c",myName[i]);
}
printf("\ncapitalized:");
gap = (int)'A' - 'a';
for(i = 0;i<8;i++)
{
    printf("%c",myName[i]+gap);
}
return 0;
}

```

lesson6_3.c

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

```

int main(void)
{
    int i,j;
    int array[2][5] = {{1,2,4,8,16},{1,3,9,27,81}};
    printf("content [0][3] = %d\n",array[0][3]); // refer one element
    printf("dump:\n");
    for(i = 0;i < 2;i++) // print all element
    {
        for(j = 0;j < 5;j++)
        {
            printf("content [%d][%d] = %d\n",i,j,array[i][j]);
        }
    }
    array[0][3] = 99; // change an element
    printf("modified array:\n"); // refer one element
    for(i = 0;i < 2;i++) // print all element
    {
        for(j = 0;j < 5;j++)
        {
            printf("content [%d][%d] = %d\n",i,j,array[i][j]);
        }
    }
    return 0;
}

```

lesson6_4.c

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

```

int main(void)
{

```

```

float num[50],sum = 0.0;
int i,count;
printf("how many numbers will you input? (max 50):");
scanf("%d" , &count);

for(i = 0;i < count; i++)
{
    scanf("%f", &num[i]);
}

printf("registered numbers\n");
for(i = 0;i < count; i++)
{
    printf("%f\n",num[i]);
    sum += num[i];
}
printf("average: %.5f", sum / (double)count);
return 0;
}

```

Preprocessor

- preprocessor will translate all statements which begins with # **before compile**
- you can define a constant number as a **capitalized text**
- you can also define **small routine** with it
- you can fork your source code using **#define #ifdef #ifndef #endif**
- you can program also easy control structure inside using **#if**
- **#undef** invalidates **#define**

lesson6_5.c

```

#include <stdio.h>
#define PI 3.14159265

int main(void)
{
    float r;
    printf("input radius:");
    scanf("%f",&r);
    printf("area = %f",PI * r * r);

    return 0;
}

```

lesson6_6.c

```

#include <stdio.h>
#define PI 3.14159265

```

```
#define CIRCLEAREA(x) (PI * x * x)

int main(void)
{
    float radius = 3;
    printf("radius:%f area:%f",radius,CIRCLEAREA(radius));
    return 0;
}
```

lesson6_7.c

```
#include <stdio.h>
#define PI 3.14159265
#define CIRCLEAREA(x) (PI * x * x)

#define DEBUG

int main(void)
{
    float radius = 3;

    #ifdef DEBUG
        printf("pi:%f radius:%f area:%f\n",PI,radius,CIRCLEAREA(radius));
    #else
        printf("%f\n",CIRCLEAREA(radius));
    #endif

    #ifndef MESSAGE
        printf("Thank you");
    #endif

    return 0;
}
```

lesson6_8.c

```
#include <stdio.h>

int main(void)
{
    #ifdef ESCAPE
        // you can not comment out whole program
        /*
            printf("test message");
        */
        printf("test message");
    #endif

    return 0;
}
```

```
}
```

lesson6_9.c

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

```
#define SAMPLINGRATE 48000
```

```
#define SMALLBUFFER 512
```

```
#define BIGBUFFER 1024
```

```
int main(void)
```

```
{
```

```
    int size;
```

```
    #if SAMPLINGRATE <= 44100
```

```
        float fftData[SMALLBUFFER];
```

```
        size = SMALLBUFFER;
```

```
    #elif SAMPLINGRATE > 44100
```

```
        float fftData[BIGBUFFER];
```

```
        size = BIGBUFFER;
```

```
    #else
```

```
        return 0;
```

```
    #endif
```

```
    printf("%d byte allocated for fft buffer", size * sizeof(float) );
```

```
    return 0;
```

```
}
```