

강의명 :프로그래밍

실습 번호 : 5

실습 제목 : functions and program structure (함수 및 프로그램 구조)

학생 이름 : 임지빈

학번 : 202211051

1 배열 프로그램 (1개 파일)

1.1

매크로 (macro) : #define ARRAY_SIZE 5

- ARRAY_SIZE 가 나올때마다 5를 대입해준다

전역 변수 (external variable) : int array[ARRAY_SIZE], int current=-1

- int array[ARRAY_SIZE] - 배열 변수

- int current = -1 - 일반 변수

1.2

void print_array(void)

```
{  
    int i;
```

```
    printf("(");  
    for (i = 0; i <= current; i++)  
        printf("%d%s", array[i], i == current ? " " : ",");  
    printf(")\n");  
}
```

출력하는 함수

int add_array(int item)

```
{  
    if (current == ARRAY_SIZE - 1)  
        return -1;
```

```
    current = current + 1;  
    array[current] = item;
```

```
    return item;  
}
```

더하는 함수

int delete_array(void)

```
{  
    int item;
```

```
    if (current == -1)  
        return -1;
```

```
    item = array[current];  
    current = current - 1;
```

```
    return item;  
}
```

삭제하는 함수

```

int main(void)
{
    int i;

    for (i = 0; i < ARRAY_SIZE + 1; i++) {
        printf("add_array(%d)=%d: ", i, add_array(i));
        print_array();
    }
    for (i = 0; i < ARRAY_SIZE + 1; i++) {
        printf("delete_array(=%d: ", delete_array());
        print_array();
    }
}

```

메인 함수

1.3

```

s2211051@oak:lab05$ gcc array-1-file.c -o array-1-file
s2211051@oak:lab05$ ./array-1-file
add_array(0)=0: (0)
add_array(1)=1: (0,1) automatic 변수
add_array(2)=2: (0,1,2)
add_array(3)=3: (0,1,2,3)
add_array(4)=4: (0,1,2,3,4)
add_array(5)=-1: (0,1,2,3,4)
delete_array(0)=4: (0,1,2,3)
delete_array(1)=3: (0,1,2)
delete_array(2)=2: (0,1) 정의 작성
delete_array(3)=1: (0)
delete_array(4)=0: (0)
delete_array(5)=-1: (0)

```

2 배열 프로그램 (4개 파일)

2.1

```
#define ARRAY_SIZE 5
```

```

void print_array(void);
int add_array(int item);
int delete_array(void);
2.2
#include "array-4-file.h"

```

```

int array[ARRAY_SIZE];
int current=-1;
2.3
#include <stdio.h>
#include "array-4-file.h"

```

```

extern int array[];
extern int current;

```

```

void print_array(void)
{

```

```

int i;
printf("");
for (i=0;i<= current; i++)
    printf("%d%s", array[i],i==current ? " " : ",");
printf("\n");
}

```

2.4

```
#include "array-4-file.h"
```

```

extern int current;
extern int array[];

```

```

int add_array(int item)
{
    if (current==ARRAY_SIZE -1)
        return -1;
    current = current +1;
    array[current] = item;

```

```

    return item;
}

```

```

int delete_array (void)
{
    int item;
    if (current== -1)
        return -1;

```

```

    item = array[current];
    current = current -1;
    return item;
}

```

2.5

```

#include <stdio.h>
#include <stdlib.h>
#include "array-4-file.h"

```

```

int main (void)
{
    int i;
    for (i=0; i < ARRAY_SIZE +1; i++) {
        printf("add_array(%d)=%d: ", i, add_array(i));
        print_array();
    }
    for (i=0; i<ARRAY_SIZE +1; i++) {
        printf("delete_array( )=%d: ", delete_array());
        print_array();
    }
}

```

2.6

```
s2211051@oak:lab05$ ^Cc array-4-file-1.c array-4-file-2.c array-4-file-3.c array-4-file-4.c -o array-4-file
s2211051@oak:lab05$ ./array-4-file
add_array(0)=0: (0)
add_array(1)=1: (0,1)
add_array(2)=2: (0,1,2)
add_array(3)=3: (0,1,2,3)
add_array(4)=4: (0,1,2,3,4)
add_array(5)=-1: (0,1,2,3,4)
delete_array(5)=5: (0,1,2,3)
delete_array(4)=4: (0,1,2)
delete_array(3)=3: (0,1)
delete_array(2)=2: (0)
delete_array(1)=1: ()
delete_array(0)=-1: ()
```

3 static automatic 변수

3.1

```
static int x=0;
```

```
return x=x+n;
```

정적변수 static을 이용하여 지역변수에서 벗어나게 해주고 다시 함수가 호출될때 초기화 무시하고 n만큼 더하여 반환한다

3.2

```
s2211051@oak:lab05$ gcc counter.c -o counter
```

```
s2211051@oak:lab05$ ./counter
```

```
counter(0)=0
```

```
counter(1)=1
```

```
counter(2)=3
```

```
counter(3)=6
```

```
counter(4)=10
```

```
counter(0)=10
```

```
counter(-1)=9
```

```
counter(-2)=7
```

```
counter(-3)=4
```

```
counter(-4)=0
```

4 재귀 함수 fib

4.1

```
if(n == 0) return 0;
```

```
else
```

```
    if(n == 1) return 1;
```

```
    else
```

```
        return fib_log(n-1) + fib_log(n-2);
```

4.2

```
s2211051@oak:lab05$ gcc fib.c -o fib
s2211051@oak:lab05$ ./fib
> fib(4)
| > fib(3)
| | > fib(2)
| | | > fib(1)
| | | < fib(1)=1
| | | > fib(0)
| | | < fib(0)=0
| | < fib(2)=1
| | > fib(1)
| | < fib(1)=1
| < fib(3)=2
| > fib(2)
| | > fib(1)
| | < fib(1)=1
| | > fib(0)
| | < fib(0)=0
| < fib(2)=1
< fib(4)=3
```

문제풀이]재귀함수의 형태로 피보나치 수열 구현하기

피보나치수열

Desperate Undergraduate! :: HW#01 : 피보나치 수열
kimlab.kr

4.3

4.4

5 매크로 sum 정의 작성

5.1

```
{\
  int i, sum=0;\
  for (i=from; i<= to; i++){\
    sum = sum +i;\
  }\
  printf("%d\n",sum);\
}
```

5.2

```
{\
  int i, sum=0;\
  for (i=1; i<= 10; i++){\
    sum = sum +i;\
  }\
  printf("%d\n",sum);\
}
```

5.3

```
[s2211051@oak:lab05$ emacs sum-macro.c
[s2211051@oak:lab05$ gcc sum-macro.c -o sum-macro
[s2211051@oak:lab05$ ./sum-macro
55
5050
0
-5050
```

6 매크로 cal 정의 작성

6.1

6.2

6.3