

# TI DSP,MCU 및 Xilinx Zynq FPGA

## 프로그래밍 전문가 과정

이름	문지희
학생 이메일	mjh8127@naver.com
날짜	2018/3/13
수업일수	14 일차
담당강사	Innova Lee(이상훈)
강사 이메일	gcccompil3r@gmail.com

배점 : 2 점

3. 단 한 번의 연산으로 대소문자 전환을 할 수 있는 연산에 대해 기술하시오.

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

```
void change(char arr[10])
{
    int i;
    for(i=0;i<10;i++)
    {
        printf("%c",arr[i]^32);
    }
}
```

```
int main(void)
{
    char arr[10];
    scanf("%s",arr);
    printf("%s\n",arr);
    change(arr);
    return 0;
}
```

배점 : 2 점

12. 아무 문자열을 calloc 을 통해 동적할당하고 그 문자열을 배열로 옮겨보라

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

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

```
void change(char *str)
{
    int i;
    char arr[10];
    for(i=0;i<10;i++)
```

```

    {
        arr[i]=str[i];
        printf("%c",arr[i]);
    }
}

```

```

int main(void)
{
    char *str=(char*)calloc(10,sizeof(char));
    scanf("%s",str);
    change(str);
    return 0;
}

```

**배점 : 3 점**

**13. 임의의 구조체를 한 개 만들고 해당 구조체에 대한 변수 2 개를 선언한 이후  
구조체의 멤버들에 적절한 값을 입력한 후 해당 값을 서로 교환해보라**

```

#include<stdio.h>
struct asdf{
int data1;
int data2;
};

```

```

int main(void)
{
int num;
struct asdf tmp;
tmp.data1=1;
tmp.data2=2;

```

```

num=tmp.data1;

```

```
tmp.data1=tmp.data2;
tmp.data2=num;
printf("%d,%d",tmp.data1,tmp.data2);

return 0;
}
```

배점 : 2 점

16. 이것이 없으면 C 언어를 사용할 수 없다.

이것은 무엇일까 ?

Main 함수

배점 : 2 점

25. 프로그램을 최적화하여 컴파일 하는 옵션을 적고

최적화하지 않고 디버깅 옵션을 넣어서 컴파일 하는 방법을 기술하시오.

최적화하여 컴파일하는 옵션

gcc -g -o -o0 디버깅파일이름 파일이름.c

최적화하지않고 컴파일하는 옵션

gcc -g -o 디버깅파일이름 파일이름.c

## -Stack 구현

배점 : 3 점

21. 난수를 활용해서 Stack 을 구성한다.

(같은 숫자가 들어가지 않게 하고 20 개 이상 넣는다

이때 들어가는 숫자는 1 ~ 50 사이의 숫자로 넣는다)

같은 숫자가 들어가지 않게 하는 함수를 구현하지 못했음 ㅠ

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

```
#include<malloc.h>
```

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

```
#include<time.h>
```

```
typedef struct __stack{
```

```
    int data;
```

```
    struct __stack *link;
```

```
}Stack;
```

```
Stack *get_node(void)
```

```
{
```

```
    Stack *tmp;
```

```
    tmp=(Stack*)malloc(sizeof(Stack));
```

```
    tmp->link=NULL;
```

```
    return tmp;
```

```
}
```

```
void push(Stack **top,int data)
```

```
{
```

```
Stack *tmp;
tmp=*top;
*top=get_node();
(*top)->data=data;
(*top)->link=tmp;
}
```

```
int pop(Stack **top)
{
    int n;
    Stack *tmp;
    tmp=*top;
    if(*top==NULL)
    {
        printf("Stack is Empty!");
        return 0;
    }

    *top=tmp->link;
    n=tmp->data;
    free(tmp);

    return n;
}
```

```
int main(void)
```

```

{
    Stack *top=NULL;
    int i;
    int arr[20];
    int bt[]={0};
    int avl[]={0};
    srand(time(NULL));
    for(i=0;i<20;i++)
    {
        arr[i]=rand()%51;
        printf("%d\\n",arr[i]);
    }

    for(i=0;i<20;i++)
    {
        push(&top,arr[i]);
        printf("%d\\n",top->data);
    }
    for(i=0;i<20;i++)
    {
        printf("data : %d\\n",pop(&top));
    }

    return 0;
}

```