

**Xilinx Zynq FPGA,TI DSP,
MCU 기반의
프로그래밍 전문가 과정**

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

학생 – 정한별
hanbulkr@gmail.com

1.

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

```
int main(void)
```

```
{
```

```
    char i=0;
```

```
    printf("대소문자를 변경해 줍니다\n");
```

```
    scanf("%c",&i);
```

```
    if(i>='a' && i<='z')
```

```
        printf("%c -> %c\n",i,i-32);
```

```
    else if(i>='A' && i<='Z')
```

```
        printf("%c -> %c\n",i,i+32);
```

```
    else
```

```
        printf("알파벳이 아닙니다!\n");
```

```
    return 0;
```

```
}
```

2.

stack 및 queue 외에 tree 라는 자료구조는 많은 데이터들을 검색에 용이하며 더 빠르게 검색하고 데이터를 삽입하고 지울 수 있다.

3.

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

```
int main(void)
```

```
{
```

```
    int x=0,t=0,len=0;
```

```
    int i[]={0};
```

```
    printf("임의의 수 x 를 입력하시오\n");
```

```
    scanf("%d",&x);
```

```
    len = x/4096;
```

```
    if(x<4096)
```

```
        printf("%d\n", x);
```

```
    else{
```

```
        while(t <= len)
```

```
        {
```

```
            if(t==0)
```

```
                printf("%d ",x%4096);
```

```
            else
```

```
                printf("%d ",4096);
```

```

        t++;

    }
    printf("\n");
}
}

```

4.
 int p[7] 은 int 형 리턴 타입의 크기가 7 인 p 라는 배열이고
 int(*p)[7] 은 배열 일곱개 짜리 묶음을 반환하는 p 라는 배열.

6.
 intel 은 쉽게 말해 컴퓨터용으로 쓰는 cpu 라고 보면 되고
 ARM 은 임베디드 시스템 또는 저전력의 적은 데이터로 동작하는 단말기, 핸드폰이나 그 외 여러 모듈 등에 사용하는
 cpu 라고 생각하면 된다.

7.
 함수를 가지고 있는 라이브러리다.

9. 레지스트> 캐시 > 메모리 > 디스크

10. stack 지역변수, heap 동적할당, data 전역변수와 static 변수 , text 머신코드

11. call jmp 등의 분기문의 경우 파이프 라인을 부수게 한다. 한마디로 중간에 push 나 pop 등이 동작하면 지연이 발생한다.

12.

13. goto 는 분기를 파이프 라인의 깨짐을 덜 해준다. 명령어의 횟 수가 다른 것들에 비해 적기 때문이다.
 그래서 속도가 좀더 빠르다고 할 수 있다. 하지만 잘못쓰면 이상한 공간의 루프에 빠질 수 있다는 단점이 있다.

14.
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

```

int main(void)
{
    char a[]="TI Cortex-R5F Safety MCU is very good to Real-Time System.";

    printf("같은 주소 값은 %p\n",strstr(a,"Safety MCU"));

    return 0 ;
}

```

19
#include<stdio.h>
#include<malloc.h>

```

#define EMPTY 0

struct node{

    int data;
    struct node *link;

};

typedef struct node Stack;

Stack *get_node()
{
    Stack *tmp;
    tmp =(Stack *)malloc(sizeof(Stack));
    tmp -> link = EMPTY;
    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)
{
    Stack *tmp;
    int num;
    tmp = *top;
    if(*top == EMPTY)
    {
        printf("Stack is empty!!!\n");
        return 0;
    }
    num = tmp ->data;
    *top=(*top)->link;
    free(tmp);
    return num;
}

int main(void)
{
    stack *top = NULL;

```

```
push(&top, 1);  
push(&top, 2);  
printf("data = %d\n", pop(&top));  
}
```

20.

while 문 or 스택

22.

전체적인 성능대비 효율이 괜찮다. 데이터의 입출력 속도 와 검색 속도가 빠르다. 전체적으로 안좋다고 할 부분이 없다.

24. 셀

25. 병렬처리 이냐 아니냐

30. 인터럽트가 발생하면 다른 것의 진행을 멈추고 그 곳이 발생한 주소번지로 강제로 돌아가 동작을 하고 돌아오는 것이다.

31 . 페이지 방법으로 가상메모리를 관리하는 방식.

53. gpio

54. 신호를 잘게 자르면

56. 푸리에급수 지트랜스폼 등을 이용하여 주파수에서 다른 상태값을 구하기 위해서.