

TI DSP,MCU 및 Xilinx Zynq FPGA

프로그래밍 전문가 과정

이름	문지희
학생 이메일	mjh8127@naver.com
날짜	2018/3/12
수업일수	13 일차
담당강사	Innova Lee(이상훈)
강사 이메일	gcccompil3r@gmail.com

목차

1. 모의고사 1 회차 (못 풀었던 것)
2. 모의고사 2 회차

1. 모의고사 1 회차

17 번 배열에 아무 문자열이나 입력받아 각 문자가 홀수인 경우에만 출력하는 프로그램을 작성하라

```
#include<stdio.h>

int main(void)
{
    int i,size;
    char arr[]="asdfjkl";
    size=sizeof(arr)/sizeof(char);
    for(i=0;i<size;i++)
    {
        //printf("%c",arr[i]);
        if(i==0)
            printf("%c",arr[i]);
        else if(i%2==0)
            printf("%c",arr[i]);
    }
    return 0;
}
```

20 번 20 개의 배열에 rand()를 이용하여 무작위로 1 ~ 100 까지의 값을 저장하는 프로그램을 작성하라.

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

int main(void)
{
    int i;
    int arr[20];
```

```

for(i=0;i<21;i++)
{
arr[i]=(rand()%100)+1;
printf("%d\\n",arr[i]);
}
return 0;
}

```

21,22 번 그리고 각 배열의 요소의 홀수들과 짝수들의 합을 구해 출력하라.
 22. 홀수들의 합과 짝수들의 합을 곱하여 출력하라.

```

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

int main(void)
{
    int i,esum=0,osum=0,mul;
    int arr[20];
    for(i=0;i<21;i++)
    {
        arr[i]=(rand()%100)+1;
        if(arr[i]%2==0)
            esum+=arr[i];
        else if(arr[i]%2==1)
            osum+=arr[i];
        printf("%d\\n",arr[i]);
        printf("%d,%d\\n",esum,osum);
    }
    mul=osum*esum;
    printf("%d",mul);
    return 0;
}

```

```
}
```

28 번 정수형 변수 2 개를 선언하고 서로 다른 값으로 초기화한다.
이후에 포인터를 사용하여 2 개의 값을 서로 교환해보라.

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

```
int main(void)
```

```
{
```

```
int n1=2,n2=3;
```

```
int *p1,*p2,*p;
```

```
p=NULL;
```

```
p1=&n1;
```

```
p2=&n2;
```

```
p=p1;
```

```
p1=p2;
```

```
p2=p;
```

```
printf("%d,%d",*p1,*p2);
```

```
return 0;
```

```
}
```

29 번

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

```
struct SS{
```

```
int data1;
```

```
int data2;
};

int main(void)
{
    struct SS m;
    int n;
    m.data1=2;
    m.data2=3;

    n=m.data1;
    m.data1=m.data2;
    m.data2=n;

    printf(" %d,%d",m.data1,m.data2);

    return 0;
}
```

34 번

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

typedef struct __stack{
    int data;
    struct __stack *link;
}Stack;

Stack *get_node()
{
    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;
    //printf("%d",(*top)->data);
}
```

```
int pop(Stack **top)
{
    int n;
    Stack *tmp;
    tmp=*top;
    if(tmp==NULL)
    {
        printf("Stack is EMPTY!");
        return 0;
    }
    n=tmp->data;
    *top=tmp->link;
    free(tmp);
    //printf("%d",n);
    return n;
}
```

```
int main(void)
```

```
{  
    Stack *top=NULL;  
    push(&top,10);  
    push(&top,20);  
    push(&top,30);  
    //pop(&top);  
  
    printf("%d\n",pop(&top));  
    printf("%d\n",pop(&top));  
    printf("%d\n",pop(&top));  
    printf("%d\n",pop(&top));  
  
    return 0;  
}
```

40 번 volatile 키워드의 용도는 무엇일까 ?

동일한 메모리를 여러 프로세스나 스레드가 사용할 때 필요. 값이 변결될 수 있으니 사용할 때 마다 다시 메모리에서 읽으라는 의미이다.

2. 모의고사 2 회차

배점 : 2 점

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

배점 : 2 점

4. Stack 과 Queue 의 차이점에 대해 작성하시오.

Stack 은 선입후출로 먼저 들어온 데이터 값이 마지막에 나오는 형태가 되고 queue 는 선입선출로 마지막에 들어온 데이터가 먼저 나가게 되는 형태이다. Main 의 구조체에 저장된 값이 stack 은 가장 뒤늦게 생성된 동적할당 변수의 주소 값이 들어가 있고 queue 는 가장 처음 생성된 동적 할당 변수의 주소 값이 들어가 있다.

배점 : 2 점

6. `int *p[4]`와 `int (*p)[4]`의 차이에 대해 기술하시오.

`Int *p[4]`는 주소를 저장 가능한 배열 p 이고 `int(*p)[4]`는 int 형 데이터를 4 개를 저장 가능한 배열을 가리키는 포인터이다.

배점 : 2 점

7. `int[4][4]`배열이 메모리 공간에 나란히 존재함을 증명하라!

각 배열의 요소의 주소 값을 출력해보면 된다.

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

```
int main(void)
```

```
{
    int i,j;
    int arr[4][4];
    for(i=0;i<4;i++)
    {
        for(j=0;j<4;j++)
            printf("%p\\n",&arr[i][j]);
    }
}
```

```
    return 0;
}
```

라는 코드를 하나 작성하여 결과 값을 출력해보면

```
0x7ffdd6166d40
0x7ffdd6166d44
0x7ffdd6166d48
0x7ffdd6166d4c
0x7ffdd6166d50
0x7ffdd6166d54
0x7ffdd6166d58
0x7ffdd6166d5c
0x7ffdd6166d60
0x7ffdd6166d64
0x7ffdd6166d68
0x7ffdd6166d6c
0x7ffdd6166d70
0x7ffdd6166d74
0x7ffdd6166d78
0x7ffdd6166d7c
```

의 결과값이 나오는데 증가하는 주소의 값이 4로 일정하며 이는 배열의 데이터공간이 나란히 존재함을 알 수 있다.

배점 : 2 점

8. int num = 7, 육중 포인터(***)를 이용해서 이를 표현하라!**

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

```
int main(void)
{
    int num=7;
```

```

int *p1=&num;
int **p2=&p1;
int ***p3=&p2;
int ****p4=&p3;
int *****p5=&p4;
int ****p6=&p5;

printf("%d",*****p6);

return 0;
}

```

배점 : 3 점

10. int arr1[3][3] = {{1, 3, 6}, {7, 1, 9}, {3, 7, 5}}과
 int arr2[3][3] = {{3, 5, 7}, {9, 1, 8}, {5, 1, 2}}중
 배열의 각 요소의 합이 가장 큰 것은 무엇인가?(프로그래밍)

```

#include<stdio.h>
int main(void)
{
    int arr1[3][3]={1,3,6},{7,1,9},{3,7,5};
    int arr2[3][3]={3,5,7},{9,1,8},{5,1,2};
    int i,j,sum1=0,sum2=0;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            sum1+=arr1[i][j];
            sum2+=arr2[i][j];
        }
    }
}

```

```

        }
    }

    printf("%d,%d",sum1,sum2);
    return 0;
}

```

배점 : 1 점

11. int 형 변수 3 개를 할당하고자 하여 아래와 같은 코드를 작성하였다.

```

int *dyn_arr = (char *)malloc(sizeof(char) * 3);에서 잘못된 부분을 정정하라
int *dyn_arr = (int *)malloc(sizeof(int) * 3);

```

배점 : 2 점

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

배점 : 2 점

16. 이것이 없으면 C 언어를 사용할 수 없다.

이것은 무엇일까 ?

헤더파일 ? #include<stdio.h>

배점 : 2 점

17. 3 을 좌측으로 3 Shift 하는 코드를 작성하고

8 을 우측으로 2 Shift 하는 코드를 작성하시오.

```

#include<stdio.h>

```

```

int main(void)
{
    int n1,n2;

```

```

n1=3<<3;
n2=8>>2;
printf("%d,%d",n1,n2);
return 0;
}

```

배점 : 2 점

20. 3 by 3 행렬의 덧셈을 계산하는 프로그램을 작성하시오.

```

#include<stdio.h>

int main(void)
{
    int arr[3][3]={1,2,3},{2,1,3},{1,2,3};
    int brr[3][3]={3,2,1},{3,2,1},{3,2,1};
    int srr[3][3];
    int i,j;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            srr[i][j]=arr[i][j]+brr[i][j];
            printf("%d\\n",srr[i][j]);
        }
    }
    return 0;
}

```

배점 : 1 점

24. Linux 에서 vim Editor 는 굉장히 사랑받는 Editor 중 하나다.

여기서 7 줄 복사해서 붙여넣고자 하는데 어떻게 해야 7 줄을 복사 붙여넣기 할 수 있는가 ?

y7y 로 복사하고 p 로 붙여넣기

배점 : 2 점

25. 프로그램을 최적화하여 컴파일 하는 옵션을 적고

최적화하지 않고 디버깅 옵션을 넣어서 컴파일 하는 방법을 기술하시오.

gcc -g 파일이름.c