TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

C 언어 모의시험 2

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

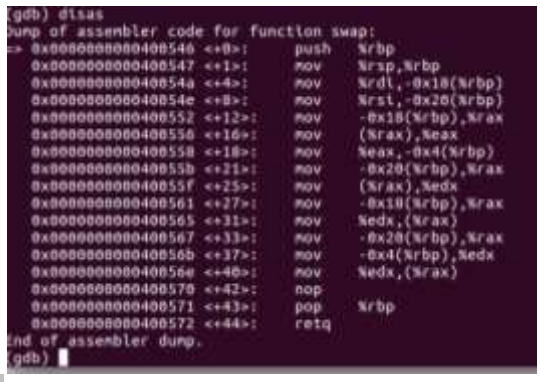
학생 - 정유경
ucong@naver.com

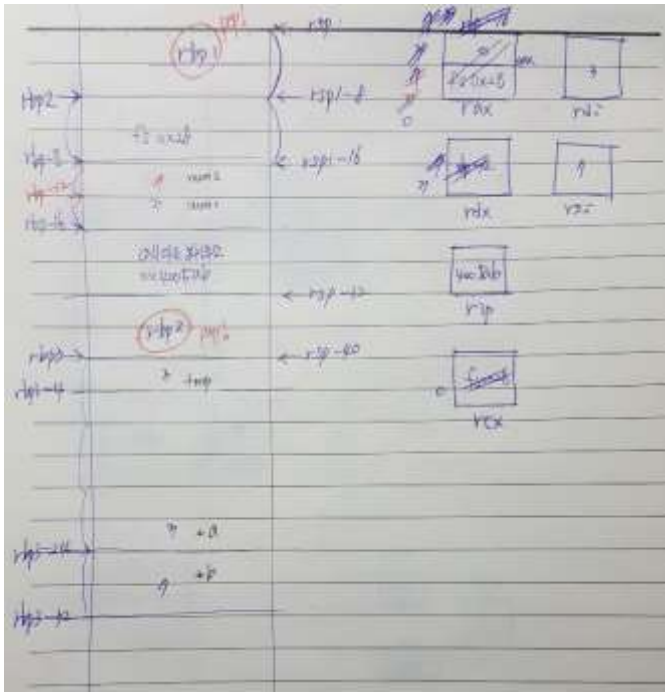
C 언어 모의시험 2 회

배점 : 20 점

`1. 아래 Code 를 작성하고 이 Code 의 기계어에 대한 그림을 그리고 분석하시오.

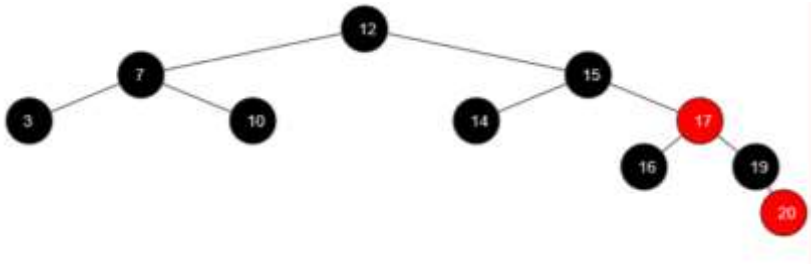
```
void swap(int *a, int *b)
{
	int tmp;
	tmp = *a;
	*a = *b;
	*b = tmp;
}

int main(void)
{
	int num1 = 3, num2 = 7;
	swap(&num1, &num2);
	return 0;
}
```





배점 : 20 점

2. **RB Tree** 에 3, 7, 10, 12, 14, 15, 16, 17, 19, 20 을 넣을 때의 그림을 그리시오.**????**



배점 : 2 점

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

1 회 중복

배점 : 2 점

4. Stack 과 Queue 의 차이점에 대해 작성하시오.

스택은 LIFO, 큐는 FIFO

배점 : 3 점

5. 임의의 값 x 가 있는데, 이를 **8192 단위????**로 정렬하고 싶다면 어떻게 해야할까 ?

(힌트 : 8192 = 2^13)

배점 : 2 점

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

1 회 중복

배점 : 2 점

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

```
#include <stdio.h>
int main()
{
	int arr[4][4]={0,};
	for(int i=0; i<4;i++)
	{
		for(int j=0;j<4;j++)
			printf("arr[%d][%d]의 주소: %p₩n",i,j,&arr[i][j]);
	}
	return 0;
}
```

```
yukyoung@yukyoung-Z20NH-AS51B1U:~/Workspace$ vi aaa.c
yukyoung@yukyoung-Z20NH-AS51B1U:~/Workspace$ gcc aaa.c
yukyoung@yukyoung-Z20NH-AS51B1U:~/Workspace$ ./a.out
arr[0][0]의 주소: 0x7ffd020526e0
arr[0][1]의 주소: 0x7ffd020526e4
arr[0][2]의 주소: 0x7ffd020526e8
arr[0][3]의 주소: 0x7ffd020526ec
arr[1][0]의 주소: 0x7ffd020526f0
arr[1][1]의 주소: 0x7ffd020526f4
arr[1][2]의 주소: 0x7ffd020526f8
arr[1][3]의 주소: 0x7ffd020526fc
arr[2][0]의 주소: 0x7ffd02052700
arr[2][1]의 주소: 0x7ffd02052704
arr[2][2]의 주소: 0x7ffd02052708
arr[2][3]의 주소: 0x7ffd0205270c
arr[3][0]의 주소: 0x7ffd02052710
arr[3][1]의 주소: 0x7ffd02052714
arr[3][2]의 주소: 0x7ffd02052718
arr[3][3]의 주소: 0x7ffd0205271c
yukyoung@yukyoung-Z20NH-AS51B1U:~/Workspace$
```

실행결과를 통해 배열요소가 연속된 주소를 가짐을 알수있다.

배점 : 2 점

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

```
#include <stdio.h>

int main()
{

int num =7;
int *p = &num;
int **pp = &p;
int ***ppp = &pp;
int ****pppp = &ppp;
int *****ppppp = &pppp;
int ******pppppp = &ppppp;

printf("%d₩n",******pppppp);

return 0;
}
```

배점 : 3 점

9. char *str = "Pointer Is Very Important"라는 문자열이 있다
여기에 대문자가 총 몇 개 사용되었는지 세어보자(프로그래밍으로)

1 회 중복

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()
```

```
{
	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 sum[3][3] = { 0, };
	int sum1 = 0, sum2 = 0;

	for (int i = 0; i < 3; i++)
	{
		for (int j = 0; j < 3; j++)
		{
			sum1+= arr1[i][j];
			sum2+= arr2[i][j];
		}
	}
	if(sum1>sum2)
		printf("배열 각 요소의 합이 가장 큰것 arr1 %d₩n",sum1);
	else if(sum1<sum2)
		printf("배열 각 요소의 합이 가장 큰것 arr2 %d₩n", sum2);
	return 0;
}
```

```
#include <stdio.h>
int main()
{
	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 sum[3][3]={0,};

	int max=0;
	int index_i, index_j;

	for(int i=0;i<3;i++)
		for(int j=0; j<3 ; j++)
			sum[i][j]=arr1[i][j]+arr2[i][j];

	for(int i=0;i<3;i++)
	{
		for(int j=0; j<3 ; j++)
		{
			if(max<sum[i][j])
			{		max = sum[i][j];

				index_i = i;
```

```
                    index_j=j;
                }
            }
        }       printf("배열 각 요소의 합이 가장 큰것: [%d 행][%d 열]=%d₩n",index_i,index_j, max);
    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

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>

int main()
{
	char *s1;
	char arr[20];
	s1 = (char*)calloc(sizeof(char), 20);
	printf("문자열을 입력하세요: ");
	scanf("%s", s1);
	for (int i = 0; i < 20; i++)
	{
		arr[i]= s1[i];
	}
	printf("문자열을 배열arr로 옮겨서 출력₩n");
	for (int i = 0; i < 20; i++)
	{
		printf("%c", arr[i]);
	}
	printf("₩n");
	free(s1);
	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

1 회 중복

배점 : 3 점

14. 7 명의 직원에 대한 급여를 입력받는다. -> 3 명으로 변경
이들이 받는 급여의 평균을 출력하고 평균 이상을 받는 사람들의 이름을 출력하라

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct Person {
	char name[20];
	int pay;
} Person;

int main()
{
	Person p[7];
	int sum = 0;
	double ave=0;
	for (int i = 0; i < 3; i++)
	{
		printf("3명의 이름과 급여를 입력합니다.");
		printf("%d번 이름 급여(만원) 순으로 입력하세요₩n",i+1);
		scanf("%s %d", p[i].name, &p[i].pay);
		sum += p[i].pay;
	}
	ave = sum / 3.0;
	printf("평균급여: %lf만원₩n", ave);
	printf("₩n<평균이상 급여자 목록>₩n");
	for (int i = 0; i<3; i++)
	{
		if (p[i].pay > ave)
			printf("%s, %d만원₩n", p[i].name, p[i].pay);
	}
	return 0;
}
```

배점 : 2 점

15. Intel Architecture 와 ARM Architecture 의 **차이점은?**
모든 프로세서는 레지스터에서 레지스터로 연산이 가능하다
인텔 x86 은 메모리에서 메모리로 연산이 가능하다
하지만 ARM 은 로드 스토어 아키텍쳐라고 하여 메모리에 있는 데이터를 처리하기 위해 메모리에서 데이터를 레지스터로 옮기고 이를 내부 프로세서에 의해 처리한 다음 이것이 다시 메모리에 쓰여진다.

배점 : 2 점
16. 이것이 없으면 C 언어를 사용할 수 없다.
이것은 무엇일까 ?
스택? 메인?

배점 : 2 점
17. 3 을 좌측으로 3 Shift 하는 코드를 작성하고
8 을 우측으로 2 Shift 하는 코드를 작성하시오.

```
#include <stdio.h>
int Rshift(int num, int x)
{
	return num >> x;
}
int Lshift(int num, int x)
{
	return num << x;
}

int main()
{
	printf("%d₩n", Lshift(3,3));
	printf("%d₩n", Rshift(8, 2));
	return 0;
}
```

배점 : 4 점
18. 1, 3, 4, 7, 11, 18, 29, 47, 76, ... 형태로 숫자가 진행된다.
40 번째 숫자는 무엇일까 ?

```
#include <stdio.h>

int fib(int n)
{
	if (n == 1)
		return 1;
```

```
        else if (n == 2)
                return 3;
        else
                return fib(n - 1) + fib(n - 2);
}

int main(void)
{
        int res;
        res = fib(40);
        printf("피보나치 40번째 숫자: %d ₩n ",res);
        return 0;
}
```

배점 : 4 점

19. 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ... 형태로 숫자가 진행된다.

1 ~ 30 번째까지의 수들로 홀수들의 합을 하고 짝수들의 합을 구한다. -> 10 번째로 변경

홀수들의 합 - 짝수들의 합의 결과를 출력하시오.

```
#include <stdio.h>

int fib(int n)
{
        if (n == 1 || n == 2)
                return 1;
        else
                return fib(n - 1) + fib(n - 2);
}

int main()
{
        int SumOdd = 0, SumEven = 0;
        int fibN = 0;
        for (int i = 0; i < 10; i++)
        {
                fibN = i + 1; // 항수
                if (fib(fibN) % 2) // 홀수
                        SumOdd += fib(fibN);
                else // 짝수
                        SumEven += fib(fibN);
        }
        printf("홀수들의 합: %d₩n", SumOdd);
        printf("짝수들의 합: %d₩n", SumEven);
        printf("홀수들의 합 - 짝수들의 합 = %d₩n", SumOdd - SumEven);
```

```
	return 0;
}
```

배점 : 2 점

20. 3 by 3 행렬의 덧셈을 계산하는 프로그램을 작성하시오.

```
#include <stdio.h>

int main()
{
	int matA[3][3] = {{1,2,3},{ 4,5,6 } ,{ 7,8,9 } };
	int matB[3][3] = { { 0,0,0 }, {1,1,1} ,{0,0,0} };
	int matSum[3][3] = { 0, };
	int i, j;

	for (i = 0; i < 3; i++)
	{
		for (j = 0; j < 3; j++)
		{
			matSum[i][j] = matA[i][j] + matB[i][j];
		}
	}
	for (i = 0; i < 3; i++)
	{
		for (j = 0; j < 3; j++)
		{
			printf("%d",matSum[i][j]);
		}
		printf("₩n");
	}
	return 0;
}
```

[복합문제]

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)

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

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

Stack* get_node()
{
	Stack *S;
	S = (Stack*)malloc(sizeof(Stack));
	//	S->data = 0;
	S->link = 0;

	return S;
}

void Push(Stack**pptop, int data)
{
	Stack *tmp;
	tmp = *pptop;
	*pptop = get_node();
	//	printf("새노드 data %d₩n",(*pptop)->data);
	//	printf("새노드 link %p₩n",(*pptop)->link);
	printf("주소: %p₩n", *pptop);
	(*pptop)->data = data;
	(*pptop)->link = tmp;
	printf("데이터: %d₩n", data);
	printf("link: %p₩n₩n", tmp);
}
int Pop(Stack**pptop)
{
	if (*pptop == 0)
	{
		printf("EMPTY!!!₩n");
		return 0;
	}

	Stack *tmp;
	tmp = *pptop;
	int pdata;
	pdata = (*pptop)->data;
	(*pptop) = (*pptop)->link;
	printf("data %d₩n", pdata);
	printf("link %p₩n₩n", *pptop);
	free(tmp);
```

```
        return pdata;
}

int main()
{
        srand(time(NULL));
        Stack *ptop = NULL;
        /*이걸 안해줘서 계속 segmentation falut 났네 */
        for (int i = 0; i < 20; i++)
                Push(&ptop, (rand()%50)+1);
        for(int i = 0; i < 22; i++)
                Pop(&ptop);
        return 0;
}
```

배점 : 5 점

22. 그 중에서 홀 수만 빼내서 AVL Tree 를 구성하도록 한다.???

스택에서 홀수인 경우만 빼내에 avl tree ins 시킨다

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

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

Stack* get_node()
{
        Stack *S;
        S = (Stack*)malloc(sizeof(Stack));
        //      S->data = 0;
        S->link = 0;

        return S;
}

void Push(Stack**pptop, int data)
{
        Stack *tmp;
        tmp = *pptop;
```

```
	*pptop = get_node();

	//	printf("새노드 data %d₩n",(*pptop)->data);
	//	printf("새노드 link %p₩n",(*pptop)->link);
	printf("주소: %p₩n", *pptop);
	(*pptop)->data = data;
	(*pptop)->link = tmp;

	printf("데이터: %d₩n", data);
	printf("link: %p₩n₩n", tmp);
}


int Pop(Stack**pptop)
{
	if (*pptop == 0)
	{
		printf("EMPTY!!!₩n");
		return 0;
	}


	Stack *tmp;
	tmp = *pptop;

	int pdata;
	pdata = (*pptop)->data;
	(*pptop) = (*pptop)->link;

	printf("data %d₩n", pdata);
	printf("link %p₩n₩n", *pptop);

	free(tmp);
	return pdata;
}




int main()
{
	int arrOdd[30] = { 0, };
```

```
	int arrEven[30] = { 0, };
	srand(time(NULL));
	Stack *ptop = NULL;
	/*이걸 안해줘서 계속 segmentation falut 났네 와... */
	for (int i = 0; i < 20; i++)
		Push(&ptop, (rand()%50)+1);

	for (int i = 0; i < 22; i++)
	{
		Pop(&ptop);
		if (Pop(&ptop) % 2) //oddnum
			arrOdd[i] = Pop(&ptop);
		else
			arrEven[i]= Pop(&ptop);
	}

	return 0;

}
```

배점 : 5 점

23. 나머지(짝 수)는 Binary Tree 로 구성한다.???

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 → p

배점 : 2 점

25. 프로그램을 최적화하여 컴파일 하는 옵션을 적고
최적화하지 않고 디버깅 옵션을 넣어서 컴파일 하는 방법을 기술하시오.

gcc -g [소스파일명] -o0 [프로그램명]
디버깅 옵션인 -g 으로 컴파일하며, 최적화 옵션은 주지 않는다 -O0