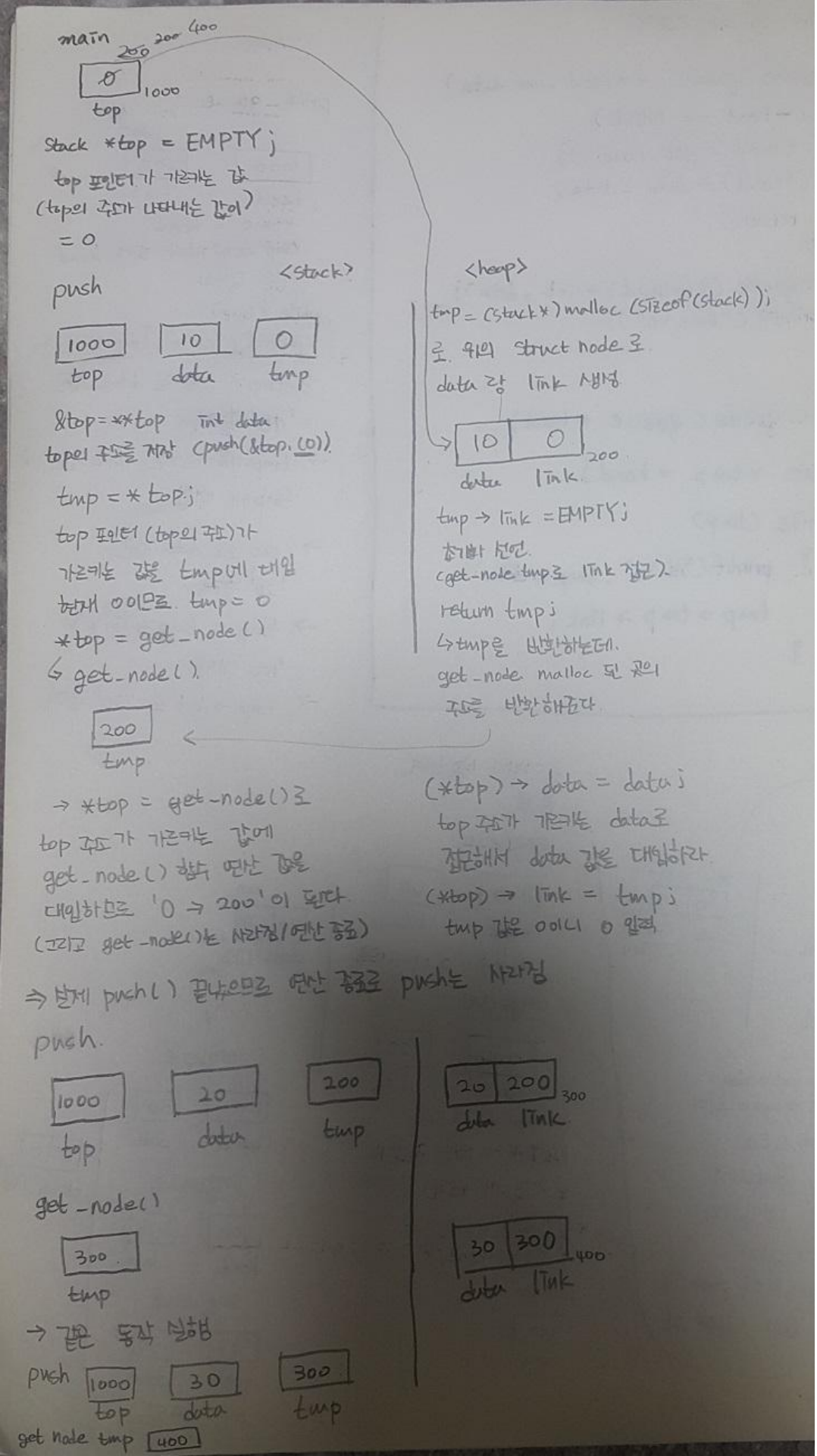
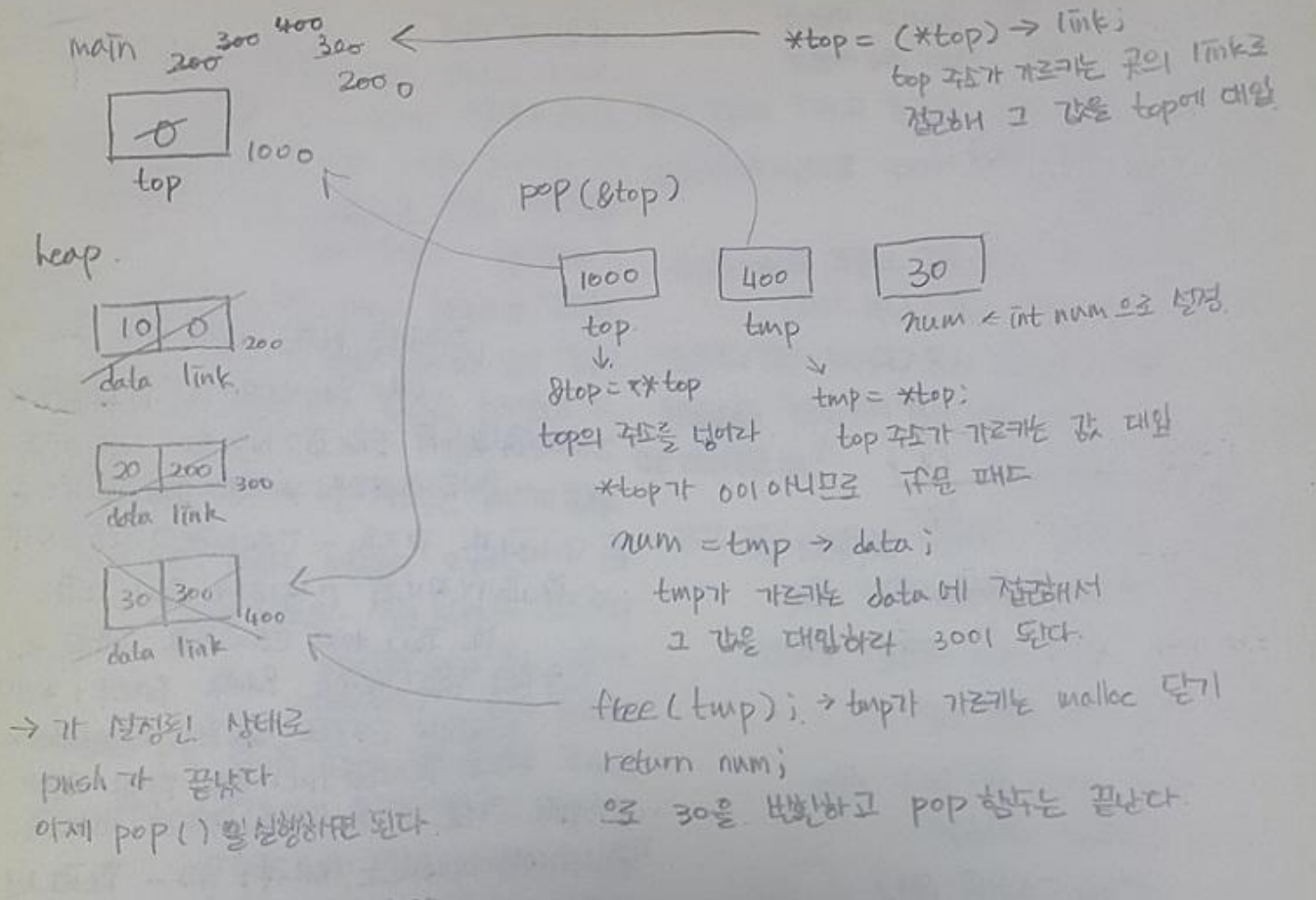
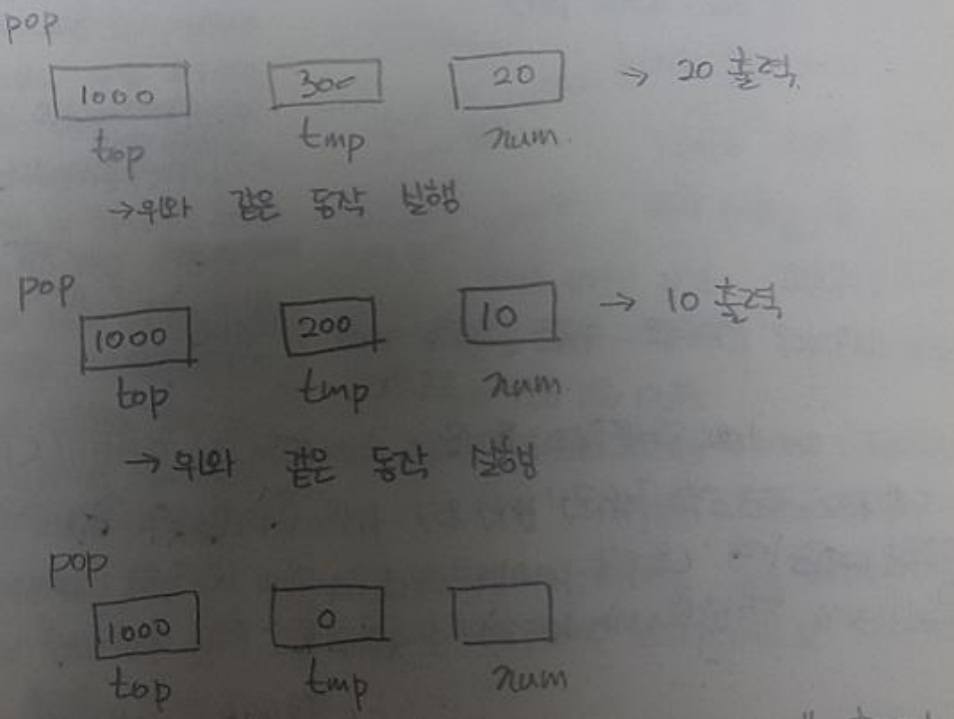
TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 : Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 : 황수정
sue100012@naver.com
8일차 (2018. 03. 05)



Stack
예시를
그림으로
표현 한 것.


main
200
300
400
300
200
0
0
1000
top
*top = (*top) -> link;
top 주소가 가르키는 곳의 link로
접근해 그 값을 top에 대입
heap
10
0
200
data link
20
200
300
data link
30
300
400
data link
pop(&top)
1000
top
400
tmp
30
num < int num으로 설정
&top = **top
top의 주소를 넘겨라
tmp = *top;
top 주소가 가르키는 값 대입
*top가 0이 아니므로 if문 패스
num = tmp -> data;
tmp가 가르키는 data에 접근해서
그 값을 대입하라 30이 된다
free(tmp); -> tmp가 가르키는 malloc 닫기
return num;
으로 30을 반환하고 pop 함수는 끝난다
-> 가 설정된 상태로
push가 끝났다
이제 pop() 을 실행하면 된다

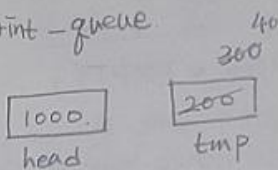

pop
1000
top
300
tmp
20
num
-> 20 출력
-> 위와 같은 동작 실행
pop
1000
top
200
tmp
10
num
-> 10 출력
-> 위와 같은 동작 실행
pop
1000
top
0
tmp
num
if문 실행됨 -> "Stack is empty!! \n" 출력됨
return 0; 이라
Stack is empty!! 후 0도 출력된다

data = 10, 20, 30 잡는다

```
Void enqueue (queue **head, int data)
{ if (*head == NULL)
  { *head = get_node();
    (*head) -> data = data;
    return;
  }
  enqueue (&(*head) -> link, data);
  printf ("test \n");
}

Void print_queue (queue *head)
{ queue *tmp = head;
  while (tmp)
  { printf ("%d \n", tmp -> data);
    tmp = tmp -> link;
  }
}
```

print_queue



*tmp = head
tmp 포인터가 가르키는 값에 head 값을 넣어라

while (tmp)

printf ("%d \n", tmp -> data);

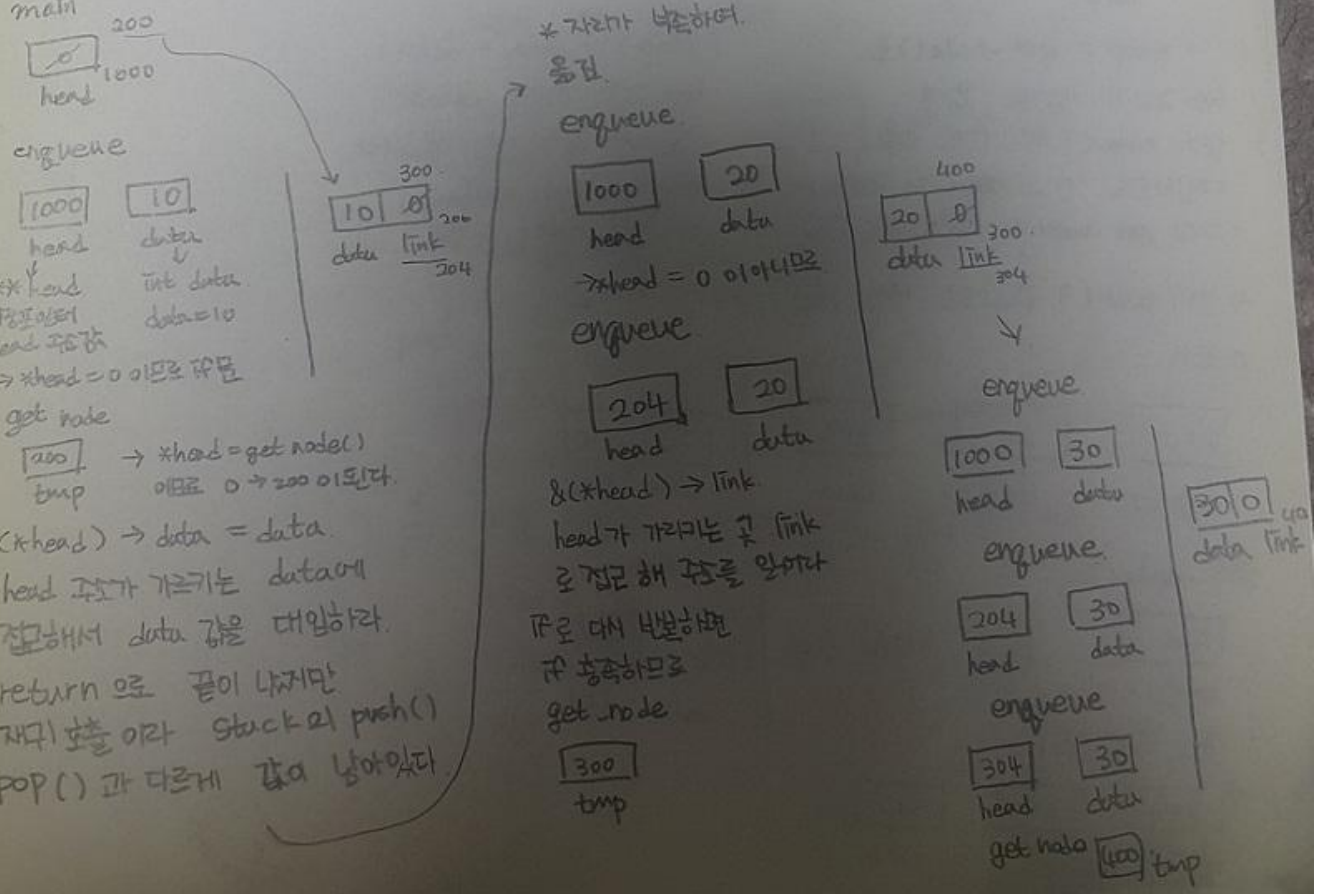
→ tmp가 가르키는 data로 접근. '10' 출력.

→ tmp가 가르키는 link 접근 tmp에 대입

→ 300 으로 data 접근 '20' 출력

→ tmp = tmp -> link 로 400 대입

→ tmp -> data 로 '30' 출력



과제2. c언어 복습 문제

1~1000 사이에 4나 6으로 나누어도 나머지가 1인 수의 합을 구하라.

```c
#include <stdio.h>

int syn(int start, int end, int t1, int t2)
{
        int res = 0, i = start;

        while(i < end + 1)
        {
           if(((i % 4) == 1) || ((i % 6) == 1))
             {
                res += i;
             }
               i++;
        }

        return res;
}

int main(void)
{
   printf("tot series sum = %d₩n", syn(1, 1000, 4, 6));
        return 0;
}
```

```c
#include <stdio.h>

int sum(int s, int e, int t1, int t2)
{
        int i, sum = 0;

        for(i = s; i <= e; i++)
                if(((i % 4) == 1) || ((i % 6) == 1))
                        sum += i;

        return sum;
}

int main(void)
{
    printf("sum of 3 series = %d₩n", sum(1, 1000, 4, 6));
        return 0;
}
```

위의 두 개의 코드는 선생님께서 각각
While과 for로 잡으신 코드이다.
< 처음 문제 풀이 당시 잡았던 코드이다. 합이 아닌 수의 값들이 출력 되었다.

```c
#include<stdio.h>
 int main(void)
 {
    int a = 1;
while (a <= 1000)
{
    if (a == 1)
  {
         a++;
  }
    else
  {
      if (a % 4 == 1)
      if (a % 6 == 1)
     printf("%d₩n", a);
           a++;
          }
                  }
}
```

```c
#include <stdio.h>

int sum(int a, int b)
{
int i, sum = 0;
for (i = a; i <= b; i++)
if (((i % 4) == 1) || ((i % 6) == 1))
sum += i;
return sum;
}

int main(void)
{
printf("1~1000 사이에 4나 6으로 나누어도
 나머지가 1인 수의 합은 %d이다₩n", sum(1, 1000));
return 0;
}
```

＜옆은 다시 코드를
잡아 본 것으로
합이 도출되어
166167 이 나왔다.