

Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

강사 - 이상훈

gcccompil3r@gmail.com

수강생 - 서재언

20640@naver.com

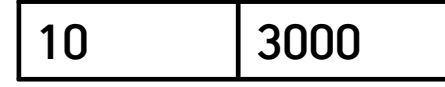
1. day_8/02.c

– Dequeue()

1000



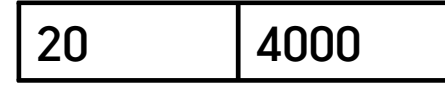
head



data

link

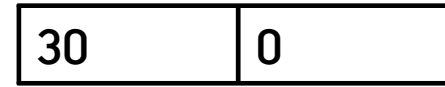
2000



data

link

3000



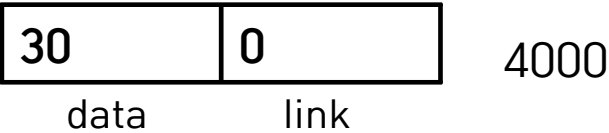
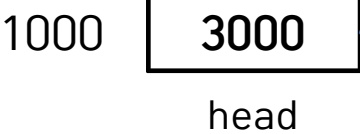
data

link

4000

1. day_8/02.c

- Dequeue()



1. day_8/02.c

– Dequeue()

1000



head



data

link

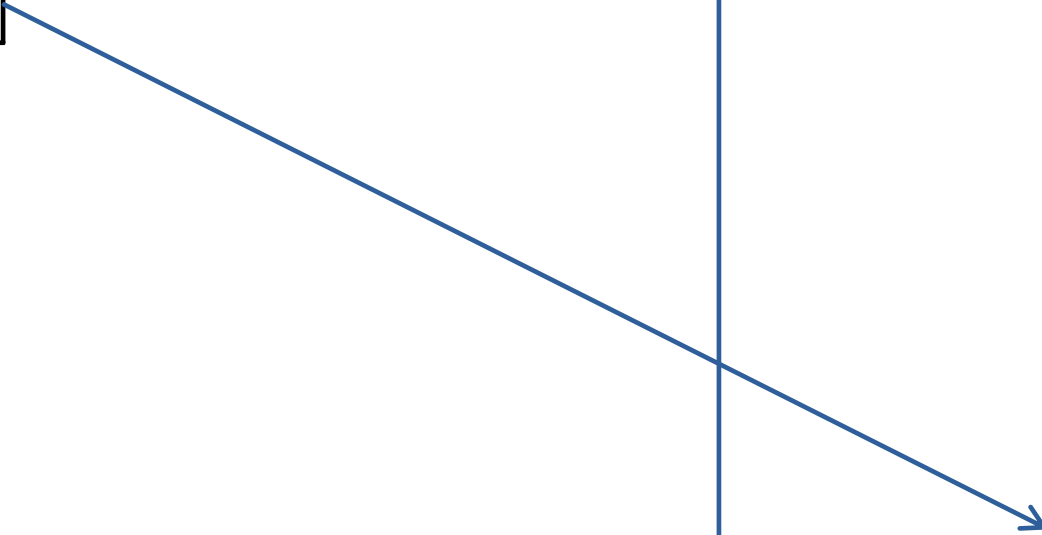
3000



data

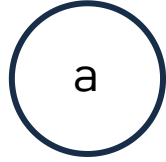
link

4000



1. day_8/03.c

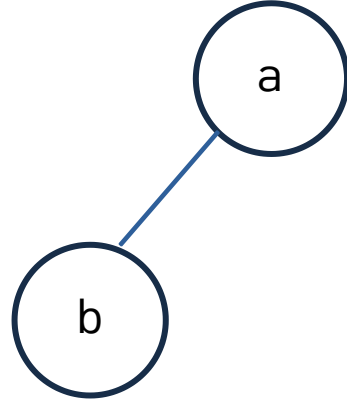
– tree_ins(root, data)



```
tree_int(tree *root, int data)
{
    if(root == NULL)
    {
        root에 메모리 생성
        root에 data 입력
    }
    .....
}
```

1. day_8/03.c

– tree_ins(root, data)

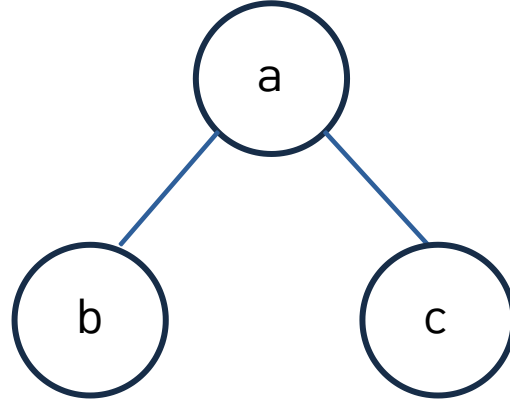


```
tree_ins(root, b)
{
    if(root == NULL)
    {
        root에 메모리 생성
        root에 data 입력
    }
    else
    {
        (1) 인자값 b가 root의 data값 a보다 작거나 같은 경우
            tree_ins(root->pLeft, b)

        (2) 인자값 b가 root의 data값 a보다 큰 경우
            tree_ins(root->pRight, b)
    }
}
```

1. day_8/03.c

– tree_ins(root, data)

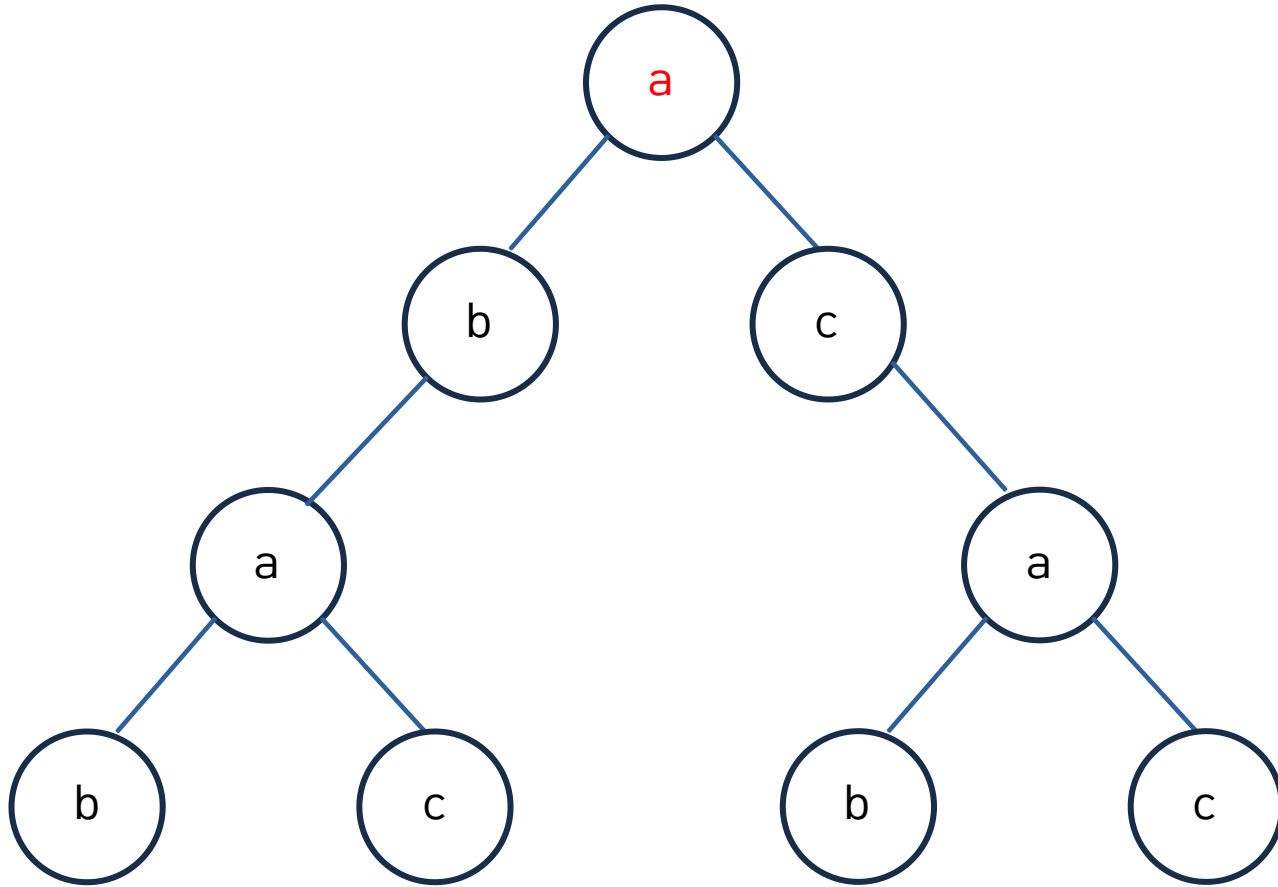


```
tree_ins(root, c)
{
    if(root == NULL)
    {
        root에 메모리 생성
        root에 data 입력
    }
    else
    {
        (1) 인자값 c가 root의 data값 a보다 작거나 같은 경우
            tree_ins(root->pLeft, c)

        (2) 인자값 c가 root의 data값 a보다 큰 경우
            tree_ins(root->pRight, c)
    }
}
```

1. day_8/03.c

– print_tree()



```
print_tree(root)
```

```
{
```

```
    (1) root값 출력
```

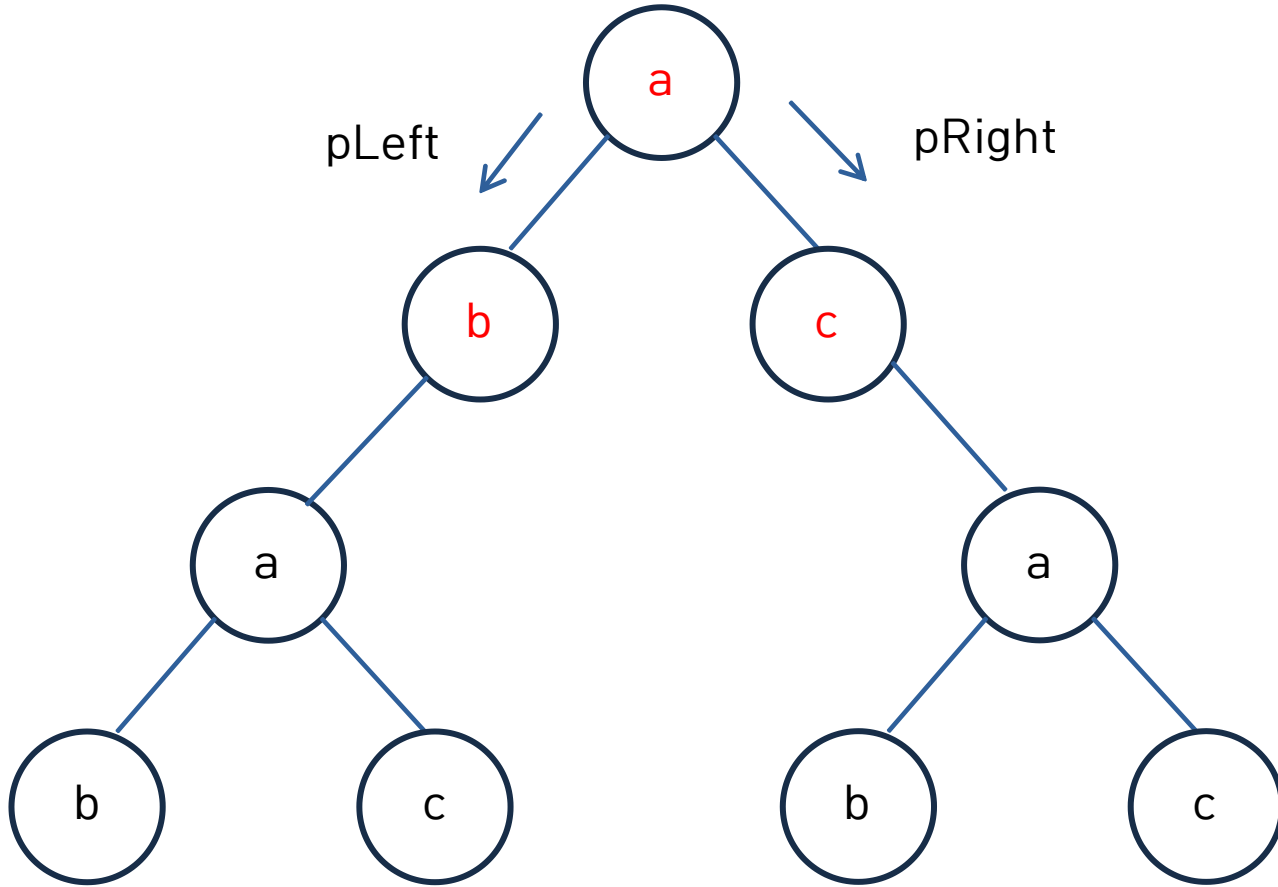
```
    (2) root->pLeft가 존재하면 pLeft의 data 출력
```

```
    (3) root->pRight가 존재하면 pRight의 data 출력
```

```
}
```


1. day_8/03.c

– print_tree()



```
print_tree(root)
```

```
{
```

```
    (1) root값 출력
```

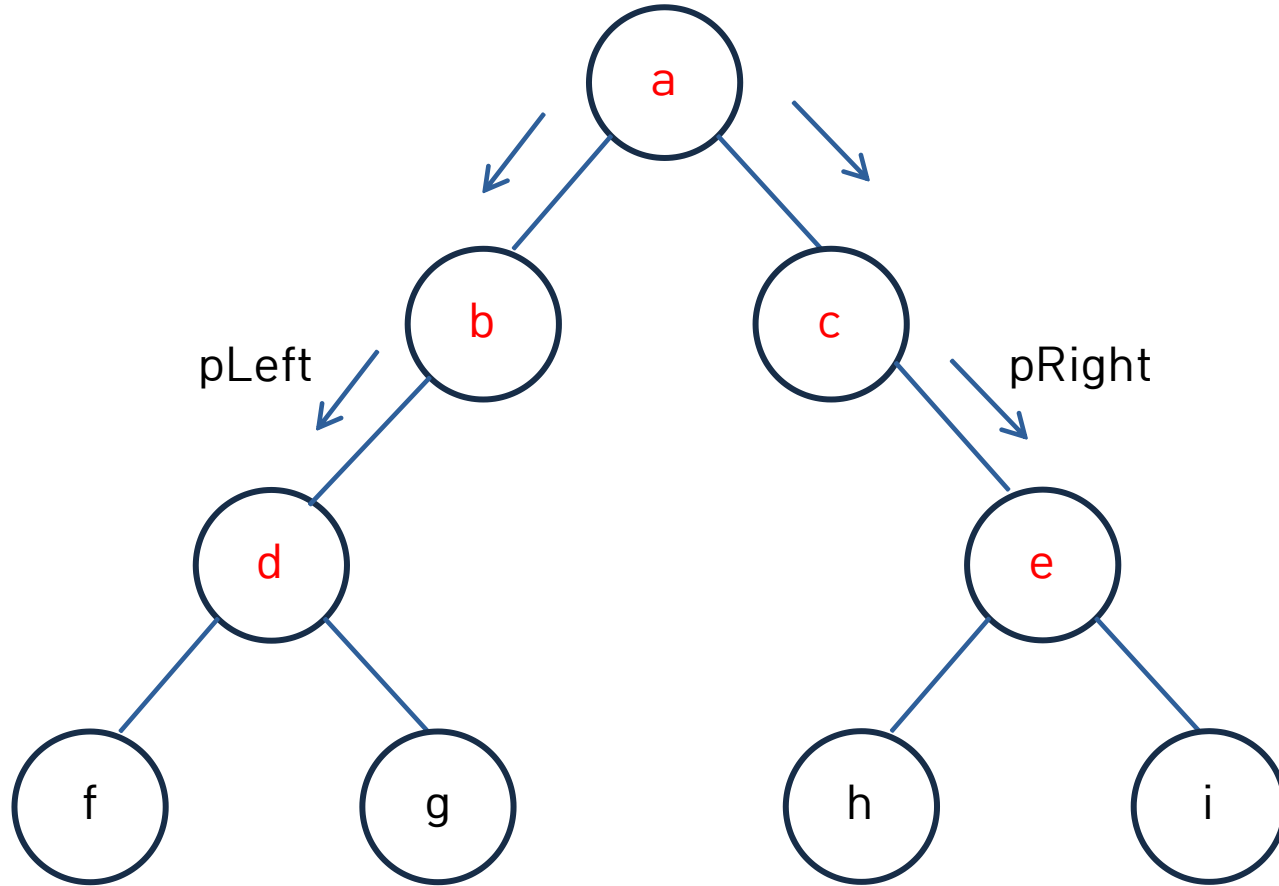
```
    (2) root->pLeft가 존재하면 pLeft의 data 출력
```

```
    (3) root->pRight가 존재하면 pRight의 data 출력
```

```
}
```

1. day_8/03.c

– print_tree()



```
print_tree(root)
```

```
{
```

```
    (1) root값 출력
```

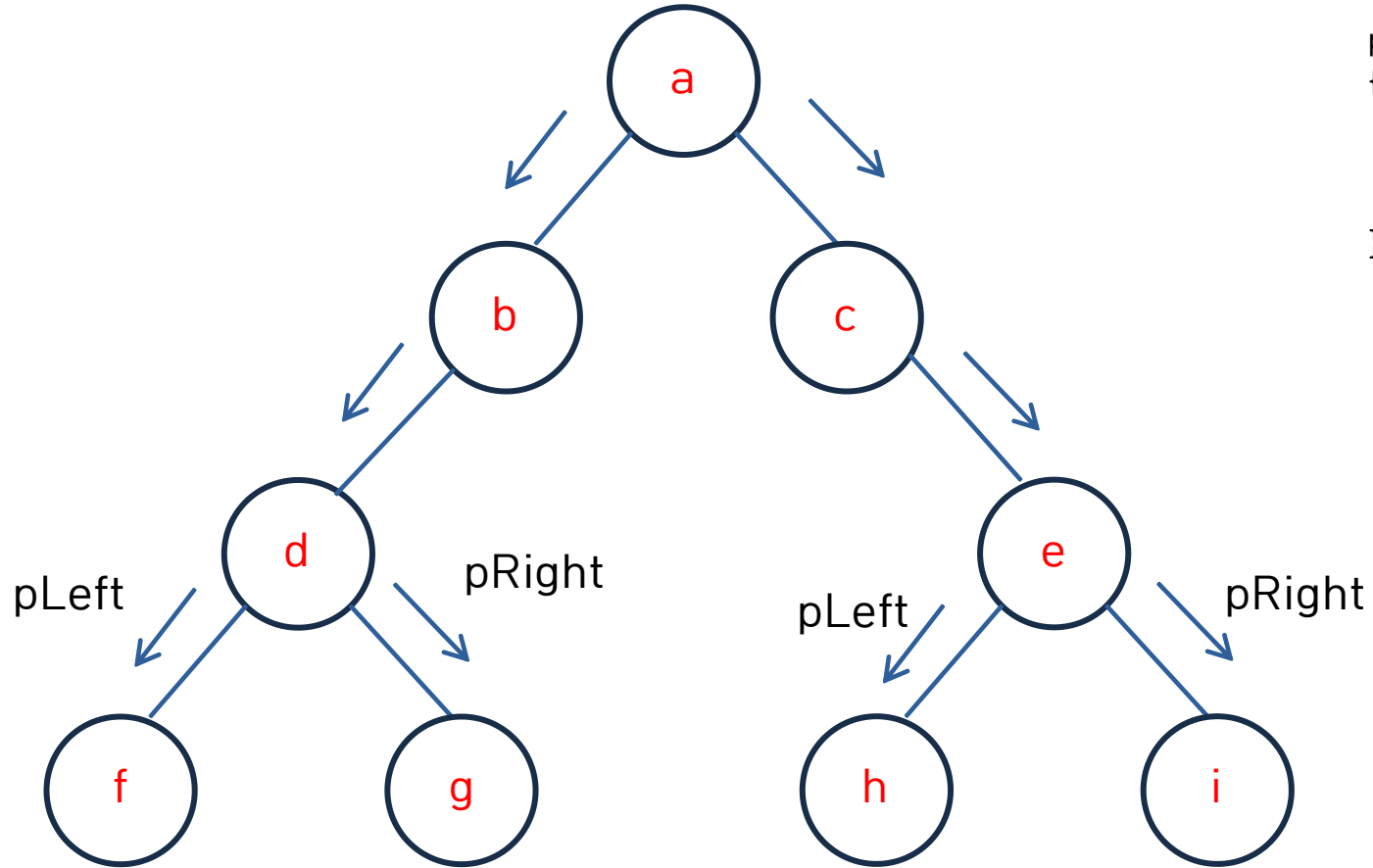
```
    (2) root->pLeft가 존재하면 pLeft의 data 출력
```

```
    (3) root->pRight가 존재하면 pRight의 data 출력
```

```
}
```

1. day_8/03.c

– print_tree()



```
print_tree(root)
```

```
{
```

```
    (1) root값 출력
```

```
    (2) root->pLeft가 존재하면 pLeft의 data 출력
```

```
    (3) root->pRight가 존재하면 pRight의 data 출력
```

```
}
```