

## <오답 정리>

### 자료구조.

#### 1.

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
//문제: 1~4096 까지 무작위로 할당되어 배열에 저장되도록 프로그래밍 하시오.
void insert(int *arr)
{
    int i;
    srand(time(NULL));
    for(i=0;i<100;i++)
        arr[i] = rand()%4096+1;
}
void print(int *arr)
{
    int i;
    for(i=1; i<=100 ;i++)
    {
        printf("%d\t",arr[i-1]);
        if((i%10) == 0)
            printf("\n");
    }
    printf("\n");
}
int main(void)
{
    int a[100]={0};

    insert(a);
    print(a);

    return 0;
}
```

#### 2.

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
//문제 2: 1~4096 까지 무작위로 할당되어 배열에 저장되도록 프로그래밍 하시오.
//각 배열은 물건을 담을 수 있는 공간에 해당한다.
//앞서서 100 개의 공간에 물건들을 담았는데 공간의 낭비가 있을 수 있다.
//이 공간의 낭비가 얼마나 발생했는지 파악하는 프로그램을 작성하시오.
int find_NULL(int *arr)
{
    int i;
    int count = 0;
    printf("check 시작\n");
    for(i=0;i<100;i++){
        if(arr[i] == '\0')
            count ++;
    }
}
```

```

        }
        return count;
    }

void insert(int *arr)
{
    int i;
    srand(time(NULL));
    for(i=0;i<100;i++)
        arr[i] = rand()%4096+1;
}

void print(int *arr)
{
    int i;
    for(i=1; i<=100 ;i++)
    {
        printf("%d\t",arr[i-1]);
        if((i%10) == 0)
            printf("\n");
    }
    printf("\n");
}

int main(void)
{
    int a[100]={0};

    insert(a);
    print(a);
    printf("낭비된 값 : %d\n",find_NULL(a));

    return 0;
}

```

### 3.

```

#include<stdio.h>
#include<time.h>
#include<stdlib.h>
int find_NULL(int *arr)
{
    int i;
    int count = 0;
    printf("check 시작\n");
    for(i=0;i<100;i++){
        if(arr[i] == '\0')
            count ++;
    }
    return count;
}

int main(void)
{
    int a[100]={0};
    int i;
    int size = 4096;
    int num=1;
    srand(time(NULL));

```

```

printf("사이즈를 선택하시오\n");
printf("사이즈는 4096 의 배수입니다.\n");
do{
printf(" 131072 까지 가능합니다. 몇배를 할지 선택해 주세요.\n");
scanf("%d",&num);
}while((size*num)>=131072);
for(i=1;i<=100;i++){
a[i-1]=rand()%(size*num)+1;
printf("%d ", a[i-1]);
if(i%10 == 0)
printf("\n");
}
printf("\n");
printf("낭비된 공간의 수는 : %d \n",find_NULL(a));
return 0;
}

```

## 4.

```

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

#define EMPTY 0

typedef struct __tree
{
    int data;
    struct __tree *link_right;
    struct __tree *link_left;
}tree;

tree *get_node()
{
    tree *tmp ;
    tmp = (tree * ) malloc(sizeof(tree));
    (tmp) → link_right = EMPTY;
    (tmp) → link_left = EMPTY;
    return tmp;
}

void binary(tree **root, int data)
{
    tree *tmp = *root;
    if(tmp == NULL){
        *root = get_node();
        (*root) → data = data;
        return ;
    }
    if((*root) → data > data)
    {
        binary(&(*root) → link_left, data);
    }
    else if((*root) → data < data)
        binary(&(*root) → link_right, data);
}

```

```

void print(tree *root)
{
    tree *tmp = root;
    if(root)
    {
        printf("data = %d", tmp -> data);

        print(root -> link_left);
        print(root -> link_right);
    }
}

int main(void)
{
    tree *root = EMPTY;
    int a[]={50,24,21,16,14,64,82,64,55,97,126,};
    int i ,num=0;
    int len = sizeof(a)/sizeof(int);

    printf("len = %d \n",len);

    for(i = 0, i<len; i++)
    {
        binary(&root, a[i]);
    }
    print(root);
    return 0;
}

```

## 5. 재귀없이 이진트리 (어려움..)

## 6. avl 재귀

## 10.

```

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

#define EMPTY 0

struct node
{
    int data;
    struct node *link;
};

typedef struct node Stack;

Stack *get_nod()
{
    Stack *tmp;
    tmp = (Stack *)malloc(sizeof(Stack));
    tmp -> link = EMPTY;
    return tmp;
}

```

```

}
void push(Stack **top, int data)
{
    Stack *tmp;
    tmp = *top;
    *top = get_nod();
    (*top) -> data = data;
    (*top) -> link = tmp;
}
int pop(Stack **top){
    Stack *tmp;
    int num;
    tmp = *top;
    if(tmp == NULL){
        printf("값이 없다\n");
        return 0;
    }
    num = tmp -> data;
    *top = (*top)->link;
    free(tmp);
    return num;
}
void make_random(Stack **top)
{
    int i;
    int arr[20]={0};
    srand(time(NULL));
    for(i=0;i<20;i++)
    {
        arr[i]=rand()%100+1;
        push(&(*top), arr[i]);
    }
}

int main(void)
{
    Stack *top;
    top = EMPTY;
    int i = 0;

    make_random(&top);

    for(i=0;i<21;i++)
        printf("%d\n",pop(&top));
    return 0;
}

```

## 17. 성적 관리 프로그램을 만들어보자.

여태까지 배운 학습 내용들을 활용하여 성적 관리 프로그램을 설계하고 구현해보자 .

**(너무 어려움)**