2018920065 루안리치 컴퓨터알고리즘 중간시험

2.

```
//
//    2018920065 luan li chi
//    midterm 21/04/21
//

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

typedef struct DListNode{
        int elem;
        struct DListNode* prev;
        struct DListNode* next;
}DListNode;

typedef struct{
        struct DListNode* H;
        struct DListNode* T;
        int size;
}SetType;

void initNode(DListNode* H, DListNode* T){
        H->next = T;
        T->prev = H;
        H->elem = T->elem = 0;
}

void initSet(SetType* S){
        S->size = 0;
        S->H = (DListNode*)malloc(sizeof(DListNode));
        S->T = (DListNode*)malloc(sizeof(DListNode));
        initNode(S->H, S->T);
}

//마지막에 노드 삽입
void addLast(SetType* S, int key){
	DListNode* n = (DListNode*)malloc(sizeof(DListNode));
	n->elem = key;
	n->prev = S->T->prev;
	S->T->prev->next = n;
	S->T->prev = n;
	n->next = S->T;
	S->size += 1;
}

//리스트 출력
void printList(SetType* S){
	DListNode* n = (DListNode*)malloc(sizeof(DListNode));
	for(n=S->H->next; n!=S->T; n=n->next){
		printf("[%d] ", n->elem);
	}
	printf("\n");
}

//리스트 생성
void setList(SetType* S, int size){
	int x;
	for(int i=0; i<size; i++){
		x=0;
		while(x>40||x<11){	//11<=x<=40
			x=(rand()%100); //랜덤 값
		}
		addLast(S, x);
	}
}

//노드의 값을 교환
void changeNode(SetType* S, DListNode* a, DListNode* b){
	DListNode* n = (DListNode*)malloc(sizeof(DListNode));
	n->elem = b->elem;
	b->elem = a->elem;
	a->elem = n->elem;
}
```

```
//노드 삭제
void deleteNode(SetType* S, DListNode* d){
	d->prev->next = d->next;
	d->next->prev = d->prev;
	S->size -= 1;
	free(d);
}

//삽입 정열
void insertSort(SetType* S){
	DListNode* n = (DListNode*)malloc(sizeof(DListNode));
	DListNode* mark = (DListNode*)malloc(sizeof(DListNode));
	mark = S->H->next->next;

	while(mark != S->T){
		n = mark;

		if(n->elem < n->prev->elem){
			changeNode(S, n, n->prev);
			n=n->prev;
		}
		else if(n->elem == n->prev->elem){
			deleteNode(S, n->prev);
		}
		else if(n->elem > n->prev->elem || n->prev == S->H){
			mark = mark->next;
		}
	}
	printList(S);
}

//선택 정열
void selectSort(SetType* S){
	DListNode* n = (DListNode*)malloc(sizeof(DListNode));
	DListNode* small = (DListNode*)malloc(sizeof(DListNode));
	DListNode* mark = (DListNode*)malloc(sizeof(DListNode));

	//initSet
```

```
    mark = S->H->next;

    while(mark != S->T){
        small = mark;
        for(n=mark->next; n!=S->T; n=n->next){
            if(n->elem < small->elem){
                small = n;
            }
            else if(n->elem == small->elem){
                n = n->prev;
                deleteNode(S, n->next);
            }
        }
        if(small != mark){
            changeNode(S, small, mark);
        }
        mark = mark->next;
    }

    //print
    printList(S);
}

//합집합
void hapjiphap(SetType* a, SetType* b){

    SetType* s = (SetType*)malloc(sizeof(SetType));
    initSet(s);
    DListNode* n = (DListNode*)malloc(sizeof(DListNode));

    //두개 리스트 합병
    for(n=a->H->next; n!=a->T; n=n->next){
        addLast(s, n->elem);
    }
    for(n=b->H->next; n!=b->T; n=n->next){
        addLast(s, n->elem);
    }
```

```
	printf("합집합:\n");
	selectSort(s); //정열
}

//차집합
void chajiphap(SetType* a, SetType* b){
	DListNode* n = (DListNode*)malloc(sizeof(DListNode));
	DListNode* m = (DListNode*)malloc(sizeof(DListNode));

	n = a->H->next; //초기화
	while(n != a->T){ //for(n = a->H->next; n != a->T; n=n->next)
		m = b->H->next; //초기화
		while(n->elem <= m->elem){
			if(n->elem == m->elem){
				n = n->prev;
				deleteNode(a, n->next);
				break;
			}
			m = m->next;
		}
		n=n->next;
	}
	printf("차집합:\n");
	printList(a);
}


int main(){

	int size;
	printf("insert size: ");
	scanf("%d",&size);
	SetType* s1 = (SetType*)malloc(sizeof(SetType));
	initSet(s1);
	SetType* s2 = (SetType*)malloc(sizeof(SetType));
	initSet(s2);

	printf("\n");
```

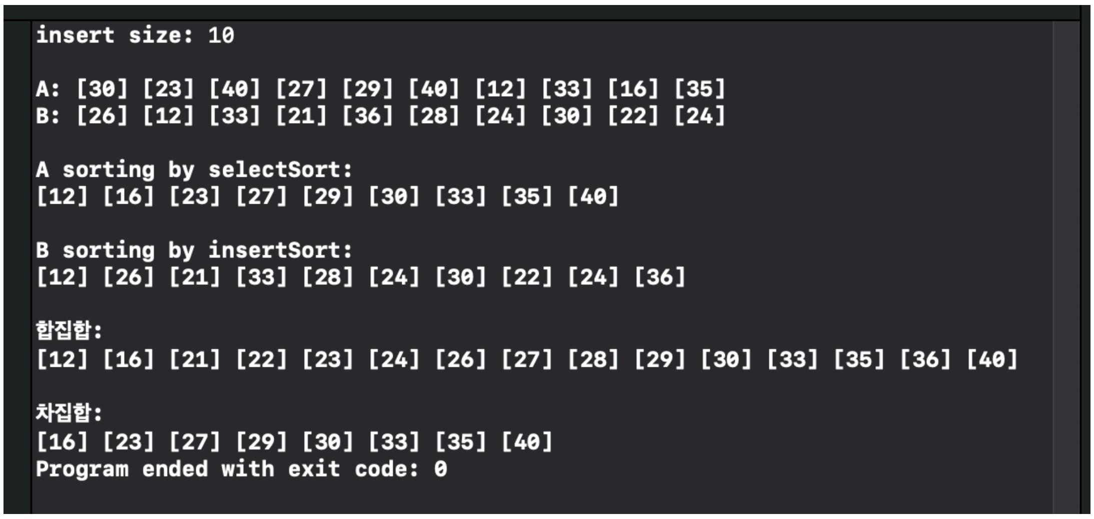
```
        setList(s1,size);
        setList(s2,size);
        printf("A: "); //출력  original a list
        printList(s1);
        printf("B: "); //출력  original b list
        printList(s2);
        printf("\n");

        printf("A sorting by selectSort:\n"); //선택정열
        selectSort(s1);
        printf("\n");
        printf("B sorting by insertSort:\n"); //삽입정열
        insertSort(s2);

        printf("\n");
        hapjiphap(s1, s2); //합집합
        printf("\n");
        chajiphap(s1, s2); //차집합

        return 0;
}
```

Simulation result:



```
insert size: 10

A: [30] [23] [40] [27] [29] [40] [12] [33] [16] [35]
B: [26] [12] [33] [21] [36] [28] [24] [30] [22] [24]

A sorting by selectSort:
[12] [16] [23] [27] [29] [30] [33] [35] [40]

B sorting by insertSort:
[12] [26] [21] [33] [28] [24] [30] [22] [24] [36]

합집합:
[12] [16] [21] [22] [23] [24] [26] [27] [28] [29] [30] [33] [35] [36] [40]

차집합:
[16] [23] [27] [29] [30] [33] [35] [40]
Program ended with exit code: 0
```