/\*------------------------------------------------------------------------------------------------

C-Programming Asignment 4-4 : 동적할당을 이용한 전화목록부 지정 프로그램.

학번 : B611019

전공 : 컴퓨터공학과 2학년

이름 : 김남일

날짜 : 2019년 06월 011일

주요 내용 :

동적할당을 활용해서 전화목록부를 생성해주는 프로그램이다. struct Node안에는 이름, 전화번호, struct pointer가 존재한다. 이름과 전화번호는 각각의 정보를 저장하고, pointer는 다음 리스트값을 가르킨다. while(1)안에서 처음에 Current = (Node\*)malloc(sizeof(struct Node));를 통해 동적할당을 받는다. 그리고 if (pStart == NULL) pStart = pEnd = Current;로 만약 list가 비어있을시 전부다 Current값으로 처음과 끝을 가리키는 포인터를 초기화해준다. 이렇게 된 다음 실행부터는 새로운 Currendt를 할당 받으면 pEnd->next = Current;(이때 Current->next = NULL;상태이다)를 통해서

pEnd가 새로운 노드의 정보를 받은 후에 pEnd = Current;를 통해서 pEnd자체가 제일 마지막에 생성된 Current값이 되게해준다. 즉 list1(pStart가 가리키는 중) list2 list3 (=pEnd)가 된다.이때

list3 ->next = NULL을 가진다.

이 프로그램에서는 3번 입력받은 후에는 생성되는 노드가 앞에 붙게 해주었다. ex) 6 5 4 1 2 3

이렇게 해주기 위해선 Current->next = pStart;를 통해서 Current가 생성되면서 제일 앞의 값을 가리키게 해주면 된다. 그리고나서 pStart = Current;를 해주면 새롭게 생성된 Curruent가 pStart가 된다. 즉 제일 앞부분이 되는것이다. 출력 할때는 Current = pStart; while (Current != NULL){printf("%s, %s \n", Current->name, Current->num);Current = Current->next;}를 통해서 Current가 pStart부터 순차적으로 pEnd까지 전부다 출력해준다. 마지막에는 Current = pStart; while (Current != NULL){pStart = Current->next; free(Current); Current = pStart;}로 전부다 동적할당을 해제시킨다. 이때 중요한 점은 free전에 pStart = Current->next를 해줌으로 Current가 삭제되기 전의 정보를 저장함으로 다음list를 이어 나가며 삭제하는 점이다.

------------------------------------------------------------------------------------------------\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct Node

{

char name[15];

char num[20];

struct Node \*next;

}Node;

int main()

{

char buffer[40] = "";

struct Node \*pStart = NULL;

struct Node \*pEnd = NULL;

struct Node \*Current = NULL;

while(1){

Current = (Node\*)malloc(sizeof(struct Node));

Current->next = NULL;

printf("이름을 입력하시오: ");

gets(buffer);

if (buffer[0] == '\0')

break;

strcpy(Current->name, buffer);

printf("전화번호를 입력하시오: ");

gets(buffer);

strcpy(Current->num, buffer);

if (pStart == NULL)

pStart = pEnd = Current;

else {

pEnd->next = Current;

pEnd = Current;

}

}

Current = pStart;

printf("\n");

while (1) {

Current = (Node\*)malloc(sizeof(struct Node));

printf("이름을 입력하시오: ");

gets(buffer);

if (buffer[0] == '\0')

break;

strcpy(Current->name, buffer);

printf("전화번호를 입력하시오: ");

gets(buffer);

strcpy(Current->num, buffer);

Current->next = pStart;

pStart = Current;

}

Current = pStart;

while ( Current != NULL){

printf("%s, %s \n", Current->name, Current->num);

Current = Current->next;

}

Current = pStart;

while (Current != NULL){

pStart = Current->next;

free(Current);

Current = pStart;

}

return 0;

}