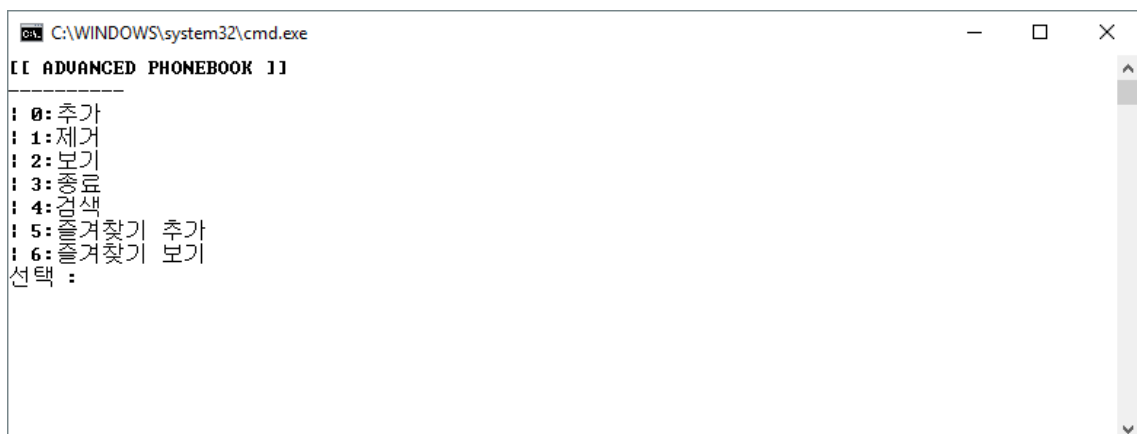


2016 프로그래밍 과제 제출

연결 리스트(Linked List)를 활용한 프로그램

작품명	Advanced Phonebook
-----	--------------------



```
C:\WINDOWS\system32\cmd.exe
[[ ADVANCED PHONEBOOK ]]
-----
! 0: 추가
! 1: 제거
! 2: 보기
! 3: 종료
! 4: 검색
! 5: 즐겨찾기 추가
! 6: 즐겨찾기 보기
선택 :
```

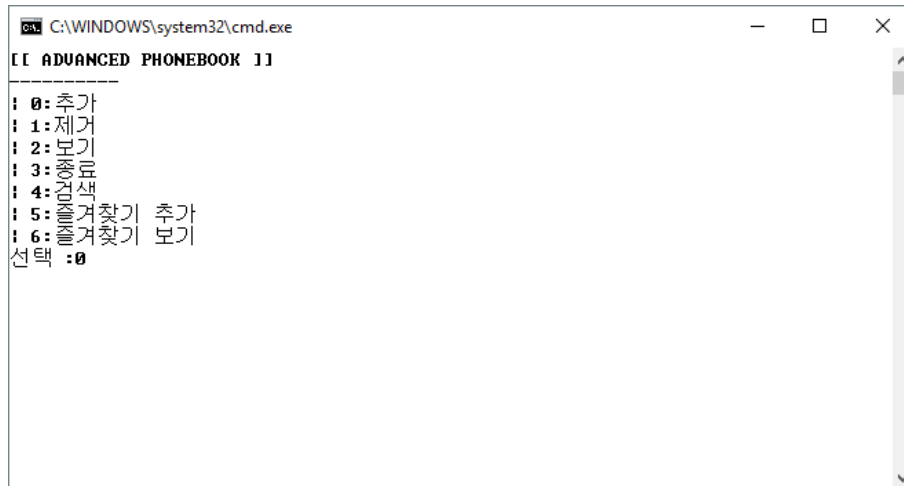
10512 김택서
10524 조나단

[본문]

I. 프로그램 소개

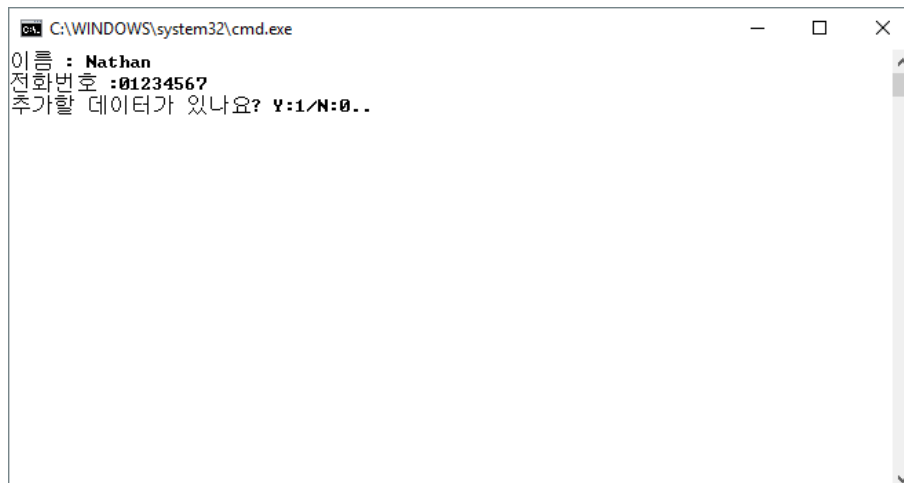
1) 프로그램의 실행화면 설명 + 주요 코드 설명

1. 추가



```
C:\WINDOWS\system32\cmd.exe
[[ ADVANCED PHONEBOOK ]]
-----
! 0: 추가
! 1: 제거
! 2: 보기
! 3: 종료
! 4: 검색
! 5: 즐겨찾기 추가
! 6: 즐겨찾기 보기
선택 : 0
```

0 을 입력해 추가 화면으로 넘어갑니다.



```
C:\WINDOWS\system32\cmd.exe
이름 : Nathan
전화번호 : 01234567
추가할 데이터가 있나요? Y:1/N:0..
```

이름과 전화번호를 입력하고 엔터를 누르면 사용자 지정 필드를 추가할지를 입력합니다.

```
C:\WINDOWS\system32\cmd.exe
이름 : Nathan
전화번호 :01234567
추가할 데이터가 있나요? Y:1/N:0..1
추가데이터 이름 : E-mail
추가할 데이터 :dev.bedrock@gmail.com
```

사용자 지정 필드의 이름과 값을 입력하고 엔터를 누릅니다.

```
C:\WINDOWS\system32\cmd.exe
이름 : Nathan
전화번호 :01234567
추가할 데이터가 있나요? Y:1/N:0..1
추가데이터 이름 : E-mail
추가할 데이터 :dev.bedrock@gmail.com

추가할 데이터가 있나요? Y:1/N:0..0
데이터가 모두 추가되었습니다.
[[ ADVANCED PHONEBOOK ]]

! 0: 추가
! 1: 제거
! 2: 보기
! 3: 종료
! 4: 검색
! 5: 즐겨찾기 추가
! 6: 즐겨찾기 보기
선택 :
```

다시 사용자 지정 필드를 생성할지를 입력하고 필드의 입력을 완료하면 다시 메인 화면으로 돌아갑니다.
사용자 지정 필드는 최대 3 개 까지 생성할 수 있습니다.

```
Node * New;
New = (Node*)malloc(sizeof(Node));
int i, j, k;
strcpy(New->NameData, name);
strcpy(New->PhonenumberData, phonenumber);
New->favorite = 0;
for (i = 0; i < 3; i++)
{
    for (j = 0; j < 2; j++)
    {
        for (k = 0; k < 30; k++)
        {
            New->custom1Data[i][j][k] = custom1[i][j][k];
        }
    }
}
New->next = NULL;

if (head == NULL)
{
    head = New;
    Last = New;
}
else
{
    Last->next = New;
    Last = New;
}
```

새 노드를 추가해서 기존 리스트가
비어있으면 새로 생성하고 존재하면 그 뒤에
붙여서 생성하는 방법을 구현했습니다.

2. 제거



```
C:\WINDOWS\system32\cmd.exe
[[ ADVANCED PHONEBOOK ]]
-----
! 0: 추가
! 1: 제거
! 2: 보기
! 3: 종료
! 4: 검색
! 5: 즐겨찾기 추가
! 6: 즐겨찾기 보기
선택 :2
```

1을 입력해 제거 화면으로 넘어갑니다.



```
C:\WINDOWS\system32\cmd.exe
삭제할 전화번호 : 01234567
```

삭제할 전화번호를 입력하고 엔터를 누릅니다.



```
C:\WINDOWS\system32\cmd.exe
보여줄 데이터가 없습니다.
[[ ADVANCED PHONEBOOK ]]
-----
! 0: 추가
! 1: 제거
! 2: 보기
! 3: 종료
! 4: 검색
! 5: 즐겨찾기 추가
! 6: 즐겨찾기 보기
선택 :
```

전화번호가 삭제된 것을 확인할 수 있습니다.

```

void Delete(char *phonenumber)
{
    if (head == NULL)
        printf("지을 데이터가 없습니다.\n");

    else
    {
        Node*temp = head->next;
        Node*tr = head;

        if (strcmp(head->PhonenumberData, phonenumber) == 0)
        {
            Node*De = head;
            head = head->next;
            free(De);
        }

        else
        {
            while (1)
            {
                if (temp == NULL)
                {
                    printf("데이터가 없습니다.\n"); break;
                }

                else if (strcmp(phonenumber, temp->PhonenumberData) == 0)
                {
                    Node*De = temp;
                    tr->next = temp->next;
                    free(De);
                    break;
                }

                else
                {
                    tr = temp;
                    temp = temp->next;
                }
            }
        }
    }
}

```

입력받은 전화번호를 가지고 있는 노드를 검색해서 그 노드의 빼고 남은 부분을 붙이고 노드의 메모리를 free 시키는 것을 구현했습니다.

3. 보기

```

C:\WINDOWS\system32\cmd.exe
[[ ADVANCED PHONEBOOK ]]
-----
! 0: 추가
! 1: 제거
! 2: 보기
! 3: 종료
! 4: 검색
! 5: 즐겨찾기 추가
! 6: 즐겨찾기 보기
선택 :2

```

2를 입력해 보기 화면으로 넘어갑니다.

```
C:\WINDOWS\system32\cmd.exe

-----
1이름 :Nathan 전화번호:01234567
1번째 사용자 데이터
E-mail
dev.bedrock@gmail.com

2번째 사용자 데이터

3번째 사용자 데이터

-----
[[ ADVANCED PHONEBOOK ]]
-----
! 0: 추가
! 1: 제거
! 2: 보기
```

입력했던 사용자 데이터를 이름과 값으로 확인 할 수 있습니다.

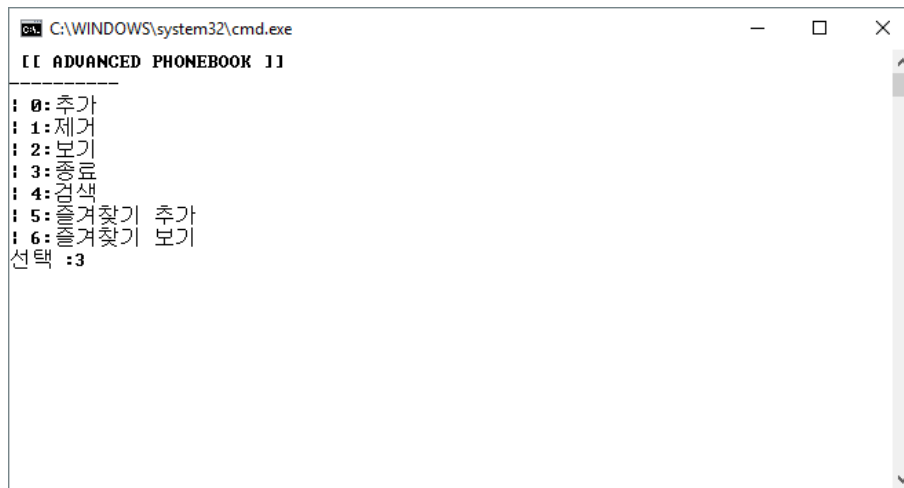
```
void Look()
{
    int i = 0, j = 0, k = 0;
    Node*temp = head;

    if (head == NULL)
    {
        printf("보여줄 데이터가 없습니다.\n");
        return;
    }

    while (1)
    {
        printf("-----\n");
        printf("1이름 :%s 전화번호:%s\n", temp->NameData, temp->PhonenumberData);
        for (i = 0; i < 3; i++)
        {
            printf("%d번째 사용자 데이터 \n", i + 1);
            if (temp->customData == 0)
            {
            }
            else
            {
                for (j = 0; j < 2; j++)
                {
                    if (temp->customData[i][j] != 0)
                        printf("%s", temp->customData[i][j]);
                    else if (temp->customData[i][j] == 0)
                    {
                        break;
                    }
                    printf("\n");
                }
            }
            printf("\n");
        }
        printf(".....\n");
        temp = temp->next;
        if (temp == NULL)
            break;
        else
            printf("\n");
    }
}
```

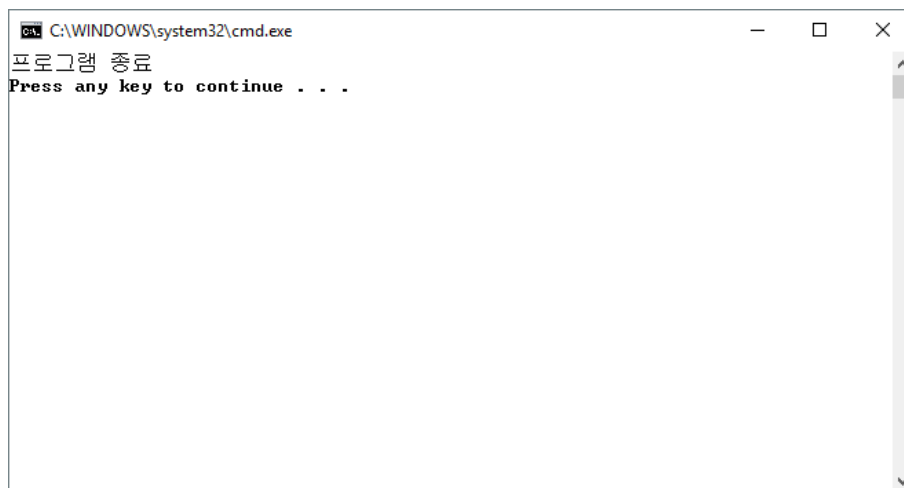
단순 반복문으로 노드에 포함된
정보들을 출력해주는 것을
구현했습니다.

4. 종료



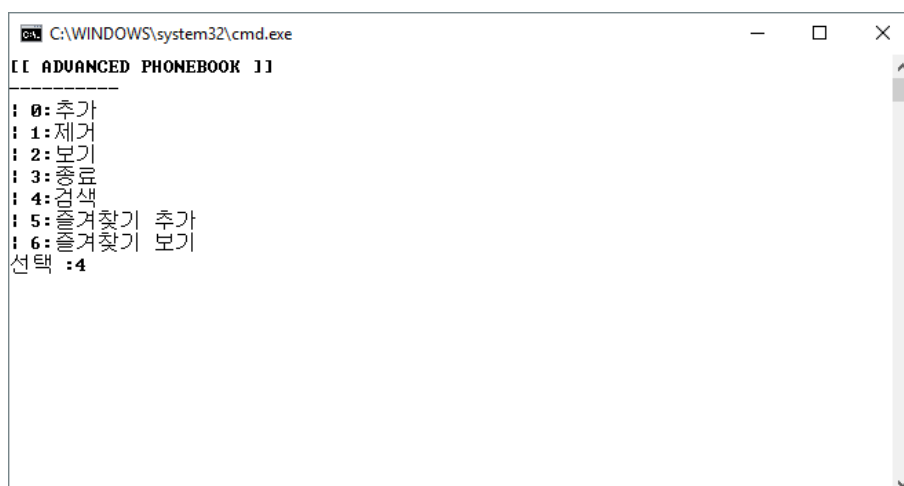
```
C:\WINDOWS\system32\cmd.exe
[[ ADVANCED PHONEBOOK ]]
-----
! 0: 추가
! 1: 제거
! 2: 보기
! 3: 종료
! 4: 검색
! 5: 즐겨찾기 추가
! 6: 즐겨찾기 보기
선택 :3
```

3을 입력해 프로그램을 종료합니다.



```
C:\WINDOWS\system32\cmd.exe
프로그램 종료
Press any key to continue . . .
```

5. 검색



```
C:\WINDOWS\system32\cmd.exe
[[ ADVANCED PHONEBOOK ]]
-----
! 0: 추가
! 1: 제거
! 2: 보기
! 3: 종료
! 4: 검색
! 5: 즐겨찾기 추가
! 6: 즐겨찾기 보기
선택 :4
```

4를 입력해 검색을 시작합니다.

```
C:\WINDOWS\system32\cmd.exe
이름을 입력해주세요:Nathan
Nathan 님의 전화번호는 01234567 입니다.
[[ ADVANCED PHONEBOOK ]]

! 0: 추가
! 1: 제거
! 2: 보기
! 3: 종료
! 4: 검색
! 5: 줄거 찾기 추가
! 6: 줄거 찾기 보기
선택 :
```

검색할 이름을 입력하고 엔터를 누르면 사용자 이름과 전화번호가 표시됩니다.

```
void Search(char * Name)
{
    Node*temp = head;

    if (head == NULL)
    {
        printf("보여줄 데이터가 없습니다.\n");
        return;
    }

    while (1)
    {
        if (strcmp(Name, temp->NameData) == 0)
        {
            printf("%s 님의 전화번호는 %s 입니다.\n",
                temp->NameData, temp->PhonenumberData);
            return;
        }

        else if (temp->next == NULL)
        {
            printf("없는 이름입니다.\n"); return;
        }

        temp = temp->next;
    }
}
```

입력받은 값을 반복문들 돌리면서 리스트의
끝까지 검색하면서 찾으면 그 값을 반환,
없으면 에러 메시지를 출력하는 것을
구현했습니다.

6. 즐겨찾기 추가

```
C:\WINDOWS\system32\cmd.exe
[[ ADVANCED PHONEBOOK ]]
-----
! 0: 추가
! 1: 제거
! 2: 보기
! 3: 종료
! 4: 검색
! 5: 즐겨찾기 추가
! 6: 즐겨찾기 보기
선택 :5
```

5를 입력해 즐겨찾기 추가 창으로 넘어갑니다.

```
C:\WINDOWS\system32\cmd.exe
즐거 찾기에 추가할 이름을 입력해주세요:Nathan
Nathan 님의 전화번호가 추가되었습니다.
[[ ADVANCED PHONEBOOK ]]
-----
! 0: 추가
! 1: 제거
! 2: 보기
! 3: 종료
! 4: 검색
! 5: 즐겨찾기 추가
! 6: 즐겨찾기 보기
선택 :
```

창에서 즐겨찾기에 추가할 사람의 이름을 입력하면 그 사람에 대한 정보가 즐겨찾기 처리됩니다.

```
void SearchF(char * Name)
{
    Node*temp = head;

    if (head == NULL)
    {
        printf("데이터가 없습니다.\n");
        return;
    }

    while (1)
    {
        if (strcmp(Name, temp->NameData) == 0)
        {
            temp->favorite;
            if (temp->favorite == 1)
                printf("이미 추가된 사람입니다.\n ");
            else
            {
                temp->favorite += 1;
                printf("%s 님의 전화번호가 추가되었습니다.\n", Name);
                return;
            }
        }

        else if (temp->next == NULL)
        {
            printf("알 수 없는 이름입니다.\n"); return;
        }
        temp = temp->next;
    }
}
```

입력받은 문자열을 기반으로 리스트의 노드를 찾아서 일치하는 노드 내부의 favorite 이라는 값을 1로 변경시켜서 즐겨찾기에 저장하는 것을 구현하였습니다.

7. 즐겨찾기 보기

```
C:\WINDOWS\system32\cmd.exe
[[ ADVANCED PHONEBOOK ]]
-----
! 0: 추가
! 1: 제거
! 2: 보기
! 3: 종료
! 4: 검색
! 5: 즐겨찾기 추가
! 6: 즐겨찾기 보기
선택 :6
```

6을 입력해 추가한 즐겨찾기를 확인하는 창으로 넘어갑니다.

```
C:\WINDOWS\system32\cmd.exe
-----
1이름 :Nathan 전화번호:01234567
1번째 사용자 데이터
E-mail
dev.bedrock@gmail.com
2번째 사용자 데이터
3번째 사용자 데이터
-----
[[ ADVANCED PHONEBOOK ]]
-----
! 0: 추가
! 1: 제거
! 2: 보기
```

이름을 입력하면 즐겨찾기에 추가된 사람들의 세부 목록이 표시됩니다.

```

else if (input == 6)
{
    system("cls");
    int i, j;
    Node*temp = head;

    if (head == NULL)
    {
        printf("보여줄 데이터가 없습니다.\n");
        return 0;
    }
    if (temp->favorite == 1)
    {
        while (1)
        {
            printf("-----\n");
            printf("1이름 :%s 전화번호:%s\n", temp->NameData, temp->PhonenumberData);
            for (i = 0; i < 3; i++)
            {
                printf("%d번째 사용자 데이터 \n", i + 1);
                if (temp->customData == 0)
                {
                    printf("데이터 없음");
                }
                else
                {
                    for (j = 0; j < 2; j++)
                    {
                        if (temp->customData[i][j] != 0)
                            printf("%s", temp->customData[i][j]);
                        else if (temp->customData[i][j][k] == 0)
                        {
                            printf("데이터 없음");
                        }
                        printf("\n");
                    }
                }
                printf("\n");
            }
            printf("-----\n");
            temp = temp->next;
            if (temp == NULL)
                break;
            else
                printf("\n");
        }
    }
}

```

전체 리스트를 반복하면서 각 노드에서 favorite의 값이 1인 노드의 정보를 모두 출력하는 것을 구현하였습니다.

II. 프로그램 제작 소감

연결 리스트가 이해하기는 쉬워도 구현하기가 어려운 개념만큼 구현하는데 어려움이 있었습니다. 연결 리스트로 구현은 했지만 기본 배열이랑 다를바가 없어서 조금 아쉬운 점도 있었습니다. 실제로 연결 리스트가 필요하고 있어야지만 효율적으로 구현할 수 있는 알고리즘들이나 모델들을 더 찾아보면서 간단하지만 중요한 개념이라는 것을 알게 되었습니다. 다음에는 좀더 개념적인 부분을 조사하고 연구해보고 싶다는 생각도 들었습니다.

https://github.com/BedrockDev/advanced_phonebook 에서 더 많은 정보를 찾아보실 수 있습니다.