

< 자료구조 > 결과 보고서

C O M E 3 3 1 0 0 6

P r o f . 김종화 교수님

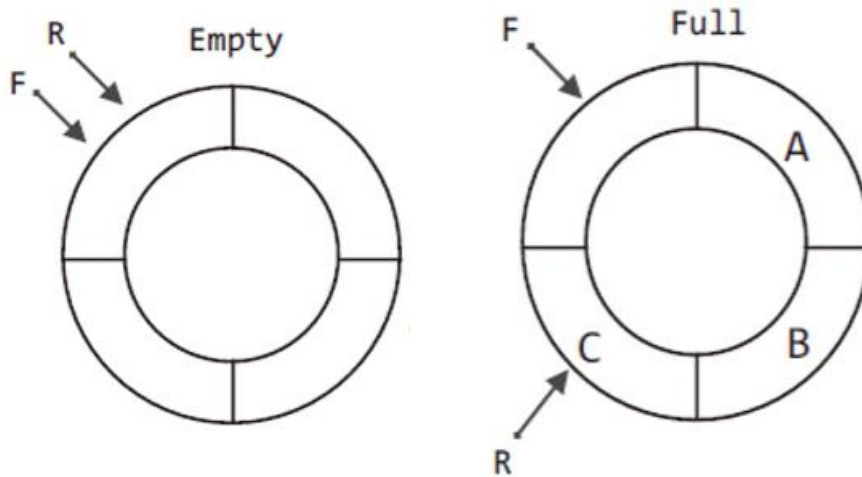
전자공학부 A

2 0 1 7 1 1 5 9 7 6 이하영

(1)circular queue를 이용한 피보나치 수열의 실행

1)간단한 설명

배열 기반 큐는 배열이 분명히 비어 있는 상태임에도 불구하고 rear포인터를 배열의 끝에서 더 이상 오른쪽으로 이동시킬 수 없기 때문에 데이터를 추가할 수 없다. 이러한 문제점을 해결하기 위해 배열의 머리와 끝을 연결한 circular(원형)구조의 queue를 사용한다.



→원형 큐에서는 Empty상태와 Full상태를 구별하기 위해 저장공간 하나를 잃는다. 이러한 성질 갖는 원형 큐를 이용하여 피보나치 수열을 저장하고(Enqueue) 반환하는(Dequeue) 코드를 살펴보자!

2)코드 분석

① circularqueueADT.h

```
typedef struct cQueue{
    int front;
    int rear;
    Data queArr[100];
}Queue;

void QueueInit(Queue* pq) {
    pq->front = 0;
    pq->rear = 0;
}

int QIsEmpty(Queue* pq) {
    if (pq->front == pq->rear)
        return 1;
    else
        return 0;
}

void Enqueue(Queue* pq, Data data) {
    if (NextPosIdx(pq->rear) == pq->front) {
        printf("Queue Memory Error!");
        exit(-1);
    }
    pq->rear = NextPosIdx(pq->rear);
    pq->queArr[pq->rear] = data;
}

Data Dequeue(Queue* pq) {
    if (QIsEmpty(pq)) {
        printf("Queue Memory Error!");
        exit(-1);
    }
    pq->front = NextPosIdx(pq->front);
    return pq->queArr[pq->front];
}

int NextPosIdx(int pos) {
    if (pos == 100 - 1)
        return 0;
    else
        return pos + 1;
}
```

→ 먼저 circularqueueADT.h에 원형 큐를 실행하기 위한 함수들을 저장해 놓았다. 위의 함수들 중에서 NextPosIdx에 의해 원형 큐의 전체적인 틀이 완성된다. 첫번째 if문이 의미하는 것은 만약 현재의 위치(pos)가 100-1과 같으면, 즉 현재 위치가 큐배열의 마지막이라면 0을 반환하고 그렇지 않으면 현재의 위치에서 오른쪽으로 한 칸 이동한 값을 반환한다. NextPosIdx함수에 의해 배열의 마지막은 비워두게 되는 원형 queue가 되는 것이다.

② 피보나치 수열 저장하고 반환하기

```
#include <stdio.h>
#include "circularqueueADT.h"
int fibo(int n);

int main() {
    Queue q;//생성
    QueueInit(&q);//초기화

    //피보나치 수열
    for (int i = 1; i < 20; i++) {
        Enqueue(&q, fibo(i));
    }

    while (!IsEmpty(&q))
        printf("%d ", Dequeue(&q));

    system("PAUSE");

    return 0;
}

int fibo(int n) {
    if (n == 1 || n == 2)
        return 1;
    else
        return fibo(n - 1) + fibo(n - 2);
}
```

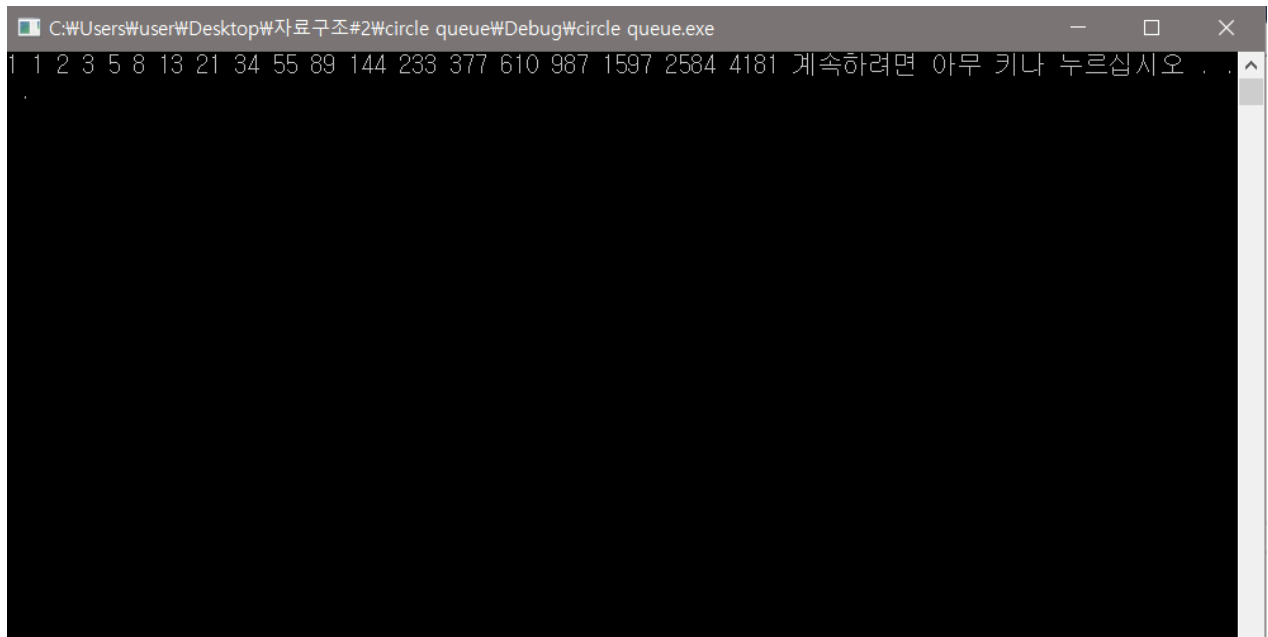
→ QUEUE구조체 변수 q를 생성하고 QueueInit함수를 통해 q의 front와 rear를 0의 값으로 초기화한다.

→ 위의 예시는 for반복문을 거치면 피보나치 수열의 1항부터 19항까지의 수들이 Enqueue를 통해 원형 큐에 값이 저장된다. 여기서 주의해야 할 점은 배열의 크기가 100이므로 저장공간을 하나 잃은 99개의 데이터만이 저장될 수 있다. 따라서 i가 1부터 시작하였으므로 100보다 크면 오류가 발생한다. 만약 for문의 범위가 (int i=1; i<100; i++)이었다면 크기가 100인 원형 큐 배열에서 저장공간을 하나 잃고 가장 많은 데이터(99개)가 저장된다.

→ Enqueue를 통해 피보나치 수열 데이터를 원형 큐에 저장하였다. 이제 Dequeue를 이용하여 원형 큐가 Empty가 아닐 때까지 데이터를 반환한다.

→ system("PAUSE");는 visual studio 2017버전을 깔았더니 cmd창이 바로 꺼져서 그것을 방지하기 위해 적어주었다!

3) 실행결과 확인하기



```
C:\Users\User\Desktop\자료구조#2\circle queue\Debug\circle queue.exe
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 계속하려면 아무 키나 누르십시오...
```

→ 피보나치 수열의 1항부터 19항까지가 출력된 것을 확인할 수 있다!

→ 더 많은 항을 보고 싶다면 for문에서 i의 범위를 바꿔주면 된다.

(1)General linear list ADT를 이용한 매장 고객 관리 리스트

1)간단한 설명

General linear list를 이용하여 매장고객의 고객등록날짜, 이름, 고객등급을 알려주는 프로그램을 작성하였다. 위의 매장고객 정보들을 CUSTOMER구조체의 멤버로 작성하고 buildList함수를 통해 만든 List의 NODE의 dataPtr부가 그 구조체를 참조하게 된다. List가 생성이 되었다면 process함수를 통해 choice에 따라 search함수와 printfList함수, insertCustomer함수가 실행된다. 따라서 미리 고객들의 정보가 저장된 메모장 파일에서 내가 원하는 고객의 정보만(search함수) 얻을 수도 있고 또는 매장전체 고객의 정보(printList함수)를 한번에 얻을 수도 있는 코드를 작성하였다. 또한 추가할 고객정보를 입력하여 효율적인 매장고객 관리 리스트를 작성하였다.

2)코드 분석

<p5-16.h>

»p5-16.h는 listLinkADT.h를 포함한다.

① COUSTOMER구조체 생성

```
typedef struct {  
    int regDate;//registrationDate 등록일  
    char name[10];  
    char customerRating[10];//customerRating 고객등급  
}CUSTOMER;
```

→구조체 멤버로 8자리수(예시:20181125)등록날짜와 고객이름, 고객등급을 갖는다. 여기서 고객등급은 3가지로 일반회원(WHITE),특별회원(SILVER),VVIP(GOLD)가 있다.

② printInstr()

```
void printInstr(void) {  
    printf("This Program prints the Customer Information\n");  
    printf("You can know customer's registrationdate,their name and their rating\n");  
    printf("the rest. Enjoy.\n");  
    return 0;  
}
```

→이 프로그램이 무엇을 실행하는지 알려주고있다. 이 프로그램을 통해 고객등록날짜와 고객이름, 고객의 등급을 알 수 있다.

③ buildList()

```
LIST* buildList(void) {
    FILE* fpData;
    LIST* list;
    int regDateIn;
    int addResult;

    CUSTOMER* pCus;

    list = createList(cmpRegistrationDate);
    if (!list) {
        printf("WaCannot create list\n");
        exit(200);
    }
    fpData = fopen("customers.txt", "r");
    if (!fpData) {
        printf("WaError opeing input file\n");
        exit(110);
    }
    while (fscanf(fpData, "%d", &regDateIn) == 1) {
        pCus = (CUSTOMER*)malloc(sizeof(CUSTOMER));

        if (!pCus) {
            printf("WaOut of Memory in build list\n");
            exit(100);
        }

        pCus->regDate = regDateIn;

        while ((fgetc(fpData)) != ' ')
            ;
        while ((fgetc(fpData)) != '"')
            ;
        fscanf(fpData, "%s", pCus->name);
        while ((fgetc(fpData)) != ' ')
            ;
        while ((fgetc(fpData)) != '"')
            ;
        fscanf(fpData, "%s", pCus->customerRating);

        addResult = addNode(list, pCus);

        if (addResult != 0) {
            if (addResult == -1) {
                printf("Memory overflow adding movie\n");
                exit(120);
            }
            else
                printf("Duplicate regDate %d not added\n", pCus->regDate);
        }
        while (fgetc(fpData) != '\n');
    }
    return list;
}
```

→createList를 통하여 list를 생성한다. 밑의 if문을 통해 list가 정상적으로 생성되었는지 확인한다.

→헤더파일 <stdio.h>에 미리 저장돼있는 구조체 유형인 FILE*로부터 fpData변수를 생성하고 "customers.txt"파일을 열고 읽기모드로 한다. 마찬가지로 if문을 통해 fpData가 정상적으로 생성되었는지 확인한다.

→fscanf를 통해 "customers.txt"파일의 수를 성공적으로 읽어 왔다면, 동적할당을 하여 CUSTOMER 구조체 포인터변수(pCus)를 생성한다. 마찬가지로 if문을 통해 동적할당이 제대로 이루어졌는지 확인한다.

→pCus의 regDate에 아까 fscanf를 통하여 읽은 regDateIn을 삽입한다.

→fgetc를 통해 fpData에서 문자 1개를 읽어온다. 만약 ' ' (띄어쓰기)이면 while문을 통과하고 ' " ' (큰따옴표)이면 while문을 통과한다. 따라서 "customers.txt"파일의 커서는 고객이름 앞에 위치하게 될 것이다.

→fscanf를 통해 fpData의 문자를 읽어오고 pCus의 name에 저장한다. 마찬가지로 위와 동일한 while문을 반복하여 고객등급을 pCus의 customerRating에 저장한다.

→addNode함수를 통하여 NODE의 dataPtr부가 구조체 포인터변수(즉 pCus)를 참조하는 NODE를 추가한다. 결국 새로 생성된 NODE는 그 NODE의 dataPtr부가 고객1의 정보를 저장하고 있는 형태일 것이다.

→fpData로부터 문자 1개를 읽어왔을 시 그 문자가 엔터이면 위의 while문을 반복하고 또 다시 새로운 NODE가 만들어지며 고객2, 고객3....의 정보를 저장 할 것이다.

④ Process(List* list)

```
void process(LIST* list) {
    char choice;

    do {
        choice = getChoice();

        switch (choice)
        {
            case 'P': printList(list);
                        break;
            case 'S': search(list);
                        break;
            case 'I': insertCustomers(list);
                        break;
            case 'Q': break;
        }
    } while (choice != 'Q');
    return;
}
```

→getChoice함수를 통해 choice를 받는다.(choice는 P,S,Q중에 하나로 getChoice함수는 뒤에서 살펴보자!)

→choice=p: printList함수가 실행되어 "customers.txt"에 저장되어 있는 모든 고객정보를 불러온다.(printList에서는 traverse함수 이용!)

→choice=S: search함수가 실행되어 날짜8자리수를 입력 받고 만약 그 날짜가 텍스트파일 안에 존재하면 구조체 pCus에 저장된 원하는 고객 정보(고객등록날짜, 고객이름, 고객등급)를 불러온다

→choice=I: insertCustomers함수가 실행되어 고객정보저장 날짜와 고객이름, 고객등급을 입력 받은 다음 fprintf를 사용하여 "customers.txt"에 세 정보를 저장한다. 즉 새로운 고객정보를 삽입한다.

→choice=Q: do-while문이 종료된다.

⑤ getChoice()

```
char getChoice(void) {
    char choice;
    bool valid;
    printf("\n===== MENU =====\n");
    printf("Here are your choices\n");
    printf(" S:Search for a registrationDate\n");
    printf(" P:Print all registrationDate\n");
    printf(" I:Insert new customers\n");
    printf(" Q:Quit\n");
    do {
        printf("=>Enter your choice:");
        scanf("%c", &choice);
        ClearLineFromReadBuffer();
        choice = toupper(choice);
        switch (choice)
        {
            case 'S': case 'P': case 'Q': case 'I':
                valid = true;
                break;
            default: valid = false;
                printf("Invalid choice, Please try again:\n");
                break;
        }
    } while (!valid);
    return choice;
}
```

→사용자로부터 문자 하나를 입력 받아 choice에 저장한다(scanf). ClearLineFromReadBuffer함수를 사용한 이유는 예를들어 choice로 P를 입력하고 엔터를 쳤을 때 printList함수가 실행되고 나서 자꾸 "Invalid choice, Please try again"문장이 떴기 때문이다. 즉 엔터도 문자로 읽어서 while문장이 실행되고, 엔터가 default값이므로 오류문구가 뜬 것이다. 따라서 ClearLineFromReadBuffer함수를 통해 엔터를 받을 때까지 입력버퍼에 남아있는 값들을 읽어주게 만들어서 정상적으로 코드가 실행되게 하였다.

⑥ Main()실행문

```
int main() {  
    LIST* list;  
  
    printfInstr();  
    list = buildList();  
    process(list);  
    printf("End Customer Information\n");  
    printf("Hope you found the customer you were looking for.");  
  
    return 0;  
}
```

→printfInstr를 통해 이 프로그래밍의 목적을 설명한다.

→buildList를 통해 "customers.txt"파일로부터 읽은 고객1,고객2,고객3,.....의 정보를 담고있는 list를 생성한다.

→process를 통해 사용자로 부터 choice문자 하나를 입력 받고 search함수, printfList함수 또는 insertCustomer함수를 실행하여 고객의 정보를 출력한다.

2) 실행결과 확인하기

- ① 고객의 정보가 들어있는 "customers.txt"파일이 존재하고 이 텍스트 문서로부터 고객의 정보를 읽어온다.

·료구조#2 > general linear list2 > general linear list2

이름	수정한 날짜	유형	크기
Debug	2018-11-25 오전 3...	파일 폴더	
customers	2018-11-24 오후 3...	텍스트 문서	1KB
general linear list2.vcxproj	2018-11-24 오전 3...	VC++ 프로젝트	7KB
general linear list2.vcxproj	2018-11-24 오전 2...	VC++ Project Filt...	2KB
general linear list2.vcxproj.user	2018-11-24 오전 1...	사용자별 프로젝...	1KB
linkListADT.h	2018-11-24 오전 2...	C/C++ 헤더	5KB
p5-16.h	2018-11-25 오전 3...	C/C++ 헤더	4KB
Test.c	2018-11-25 오전 3...	C 소스	1KB

- ② "customers.txt"파일에 들어있는 고객정보이다. 순서대로 고객등록날짜, 고객이름, 고객등급을 나타낸다.

```
customers - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
20181110 "이하영" "일반회원(WHITE)"
20181027 "홍길동" "특별회원(SILVER)"
20180415 "허균" "VVIP(GOLD)"
```

- ③ choice로 'P'를 입력하였을 때 저장되어 있던 모드 고객의 정보를 반환한다. 마지막은 'Q'를 입력하여 프로그램을 종료하였다.

```
Microsoft Visual Studio 디버그 콘솔
This Program prints the Customer Information
You can know customer's registrationdate,their name and their rating
the rest. Enjoy.

===== MENU =====
Here are your choices
  S:Search for a registrationDate
  P:Print all registrationDate
  I:Insert new cutomers
  Q:Quit
=>Enter your choice:p

<Customer Information List>
20180415 허균"      VVIP(GOLD)"
20181027 홍길동"    특별회원(SILVER)"
20181110 이하영"    일반회원(WHITE)"
End of Customer Information

===== MENU =====
Here are your choices
  S:Search for a registrationDate
  P:Print all registrationDate
  I:Insert new cutomers
  Q:Quit
=>Enter your choice:q
End Customer Information
Hope you found the customer you were looking for.
C:\Users\User\Desktop\자료구조#2\generalADT\Debug\generalADT.exe(15952 프로세스)이(가)
습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

- ④ choice로 'S'를 입력하고 고객등록일 8숫자를 입력하였지만 고객정보가 없는 날짜를 입력하였을 경우 "Sorry, but 20181128 is not available."라는 경고문구가 뜬다.

```
C:\Users\User\Desktop\자료구조#2\generalADT\Debug\generalADT.exe
This Program prints the Customer Information
You can know customer's registrationdate,their name and their rating
the rest. Enjoy.

===== MENU =====
Here are your choices
  S:Search for a registrationDate
  P:Print all registrationDate
  I:Insert new cutomers
  Q:Quit
=>Enter your choice:s
Enter a eight digit regDate: 20181128
Sorry, but 20181128 is not available.
```

- ⑤ 위의 4번의 경우에서 다시 choice로 'S'를 입력하고 올바르게 저장되어 있는 고객정보를 입력하였을 경우 고객등록날짜, 고객이름, 고객등급이 출력된다. 마지막은 'Q'를 입력하여 프로그램을 종료하였다.

```
This Program prints the Customer Information
You can know customer's registrationdate,their name and their rating
the rest. Enjoy.

===== MENU =====
Here are your choices
  S:Search for a registrationDate
  P:Print all registrationDate
  I:Insert new cutomers
  Q:Quit
=>Enter your choice:s
Enter a eight digit regDate: 20181128
Sorry, but 20181128 is not available.

===== MENU =====
Here are your choices
  S:Search for a registrationDate
  P:Print all registrationDate
  I:Insert new cutomers
  Q:Quit
=>Enter your choice:s
Enter a eight digit regDate: 20181110
20181110 이하영"    일반회원(WHITE)"

===== MENU =====
Here are your choices
  S:Search for a registrationDate
  P:Print all registrationDate
  I:Insert new cutomers
  Q:Quit
=>Enter your choice:q
End Customer Information
Hope you found the customer you were looking for.
C:\Users\User\Desktop\자료구조#2\generalADT\Debug\generalADT.exe(11464 프로세스)이(가) 0 코드로
인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

⑥ 새로운 고객정보를 입력하기 위해 choice로 'I'를 입력하였다.

```
This Program prints the Customer Information
You can know customer's registrationdate,their name and their rating
the rest. Enjoy.

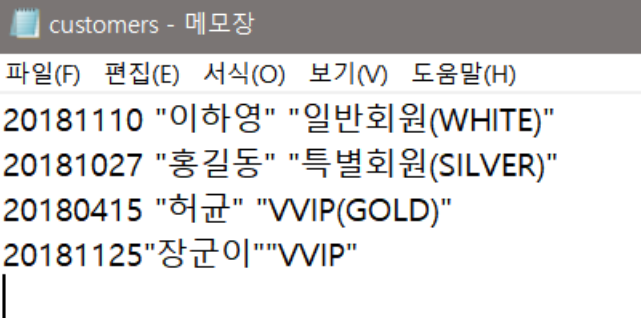
===== MENU =====
Here are your choices
  S:Search for a registrationDate
  P:Print all registrationDate
  I:Insert new cutomers
  Q:Quit
=>Enter your choice:i
고객 정보를 저장할 날짜:20181125
저장할 고객 이름:"장군이"
고객등급(일반회원,특별회원,WIP):"WIP"
고객의 정보가 저장되었습니다.

===== MENU =====
Here are your choices
  S:Search for a registrationDate
  P:Print all registrationDate
  I:Insert new cutomers
  Q:Quit
=>Enter your choice:p

<Customer Information List>
20180415 허균"      WIP(GOLD)"
20181027 홍길동"    특별회원(SILVER)"
20181110 이하영"    일반회원(WHITE)"
20181125 "장군이"  "WIP"
End of Customer Information

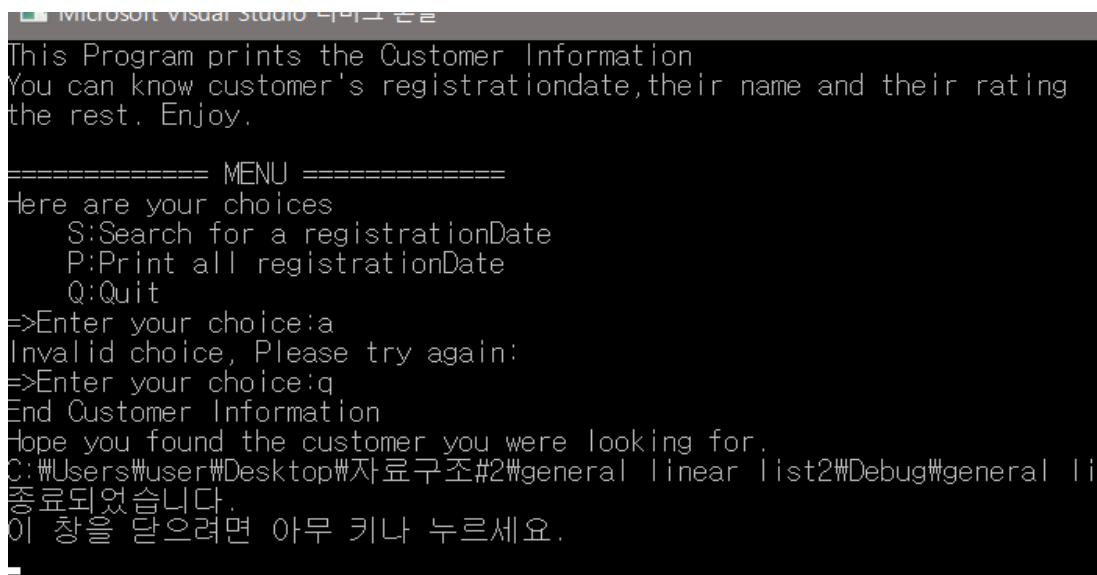
===== MENU =====
Here are your choices
  S:Search for a registrationDate
  P:Print all registrationDate
  I:Insert new cutomers
  Q:Quit
=>Enter your choice:q
End Customer Information
Hope you found the customer you were looking for.
C:\Users\user\Desktop\자료구조#2\generalADT\Debug\generalADT.exe(833
프로세스)이(가) 0 코드로 인해 종료되었습니다.
```

→날짜와 이름, 고객등급을 각각 입력하면 "customers.txt"에 장군이 고객이 삽입된다. 실제로 텍스트 파일에 저장되었는지 확인하기 위해서 바로 다음실행으로 'P'를 입력하였다. 전체명단을 확인해 보면 장군이 고객이 추가된 것을 확인할 수 있다. 마지막은 'Q'를 입력하여 프로그램을 종료하였다.



→실제로 "customers.txt"파일에 장군이 고객 정보가 등록되었음을 확인할 수 있다.

- ⑦ 'S', 'P', 'Q', 'I' 이외의 문자를 입력하였을 경우 "Invalid choice, Please try again"이라는 경고 문경고 뜨는 것을 확인할 수 있다. 마지막은 'Q'를 입력하여 프로그램을 종료하였다.



<느낀점>

저번 과제를 통해서 stack구조와 응용을 배웠습니다. 처음에 stack구조를 배울 때 포인터, 구조체 등 개념이 어려워서 고생했던 기억이 있는데 그때 한번 고생을 하고 나니 queue와 list를 이해하기가 조금이나마(아주 조금...!) 수월했습니다. 데이터를 저장하는 방법으로 C프로그래밍 할 때에는 주로 배열을 사용했었는데 자료구조를 통해 새로운 데이터 저장 방법인 원형 queue list를 알게 되었습니다. 또한 general linear list구조를 통해 데이터를 중간에 삽입하고 삭제하는 방법을 알게 되었고 serach함수를 이용하여 list에 내가 찾고자 하는 data를 발견하는 방법을 배웠습니다. 이번 과제를 하면서 C프로그래밍에서 약했던 부분인 파일처리에 대해서 깊게 공부하게 되었습니다. 이론을 배우고 나서 이번 과제를 하니깐 조금씩 이해가 되는 것 같아서 프로그래밍하는게 재밌었습니다.