

Computer Algorithm Term Project

01반 9조

2019112116 조민우

2020112097 정윤지

1. 과제 설명

20명의 Member에게는 20개의 item에 대한 고유의 선호도가 있습니다.

한 Member 에게 하나의 item 이 매칭되어야 하고, item 중복은 없어야 합니다.

각 item에 대한 Member의 선호도는 해당 item과 매칭 되었을 때의 Satisfaction으로 이어져 Satisfaction Score가 됩니다.

따라서 각 Member들의 Satisfaction의 합이 max가 될 때, 각 Member에게 배정된 item 과 Score, 그리고 Score들의 합을 출력합니다.

Random 함수를 이용해 랜덤의 Member를 선택하고, 그 Member의 Satisfaction Score가 최대인 item을 매칭하고, 이 과정을 모든 Member들이 Item을 가질 때 까지 진행합니다. 한 번 선택된 Member와 Item은 배열에 저장하여 다시 선택되지 않도록 합니다. 만약 Member가 선택할 수 있는 Item이 없을 시에 초기화를 해준 후 처음부터 다시 실행합니다.

2. Source Code 설명

<변수 설명>

Satisfaction 구조체를 선언해 줍니다.

각각의 정수형 member, item, satisfaction score 을 저장해주기 위한 변수를 선언해줍니다.

```
typedef struct Satisfaction{  
    int member;  
    int item;  
    int score;  
}Satisfaction;
```

List : 파일의 내용을 저장 할 구조체 배열

store : member 에 따른 item 을 선택할 때 순서대로 저장할 구조체 배열

result : 최종적으로 선택된 member, item, score 을 출력하기 위해 저장할 구조체 배열

check_items : 선택된 item 의 방문처리를 위한 배열

check_member : 선택된 member 의 방문처리를 위한 배열

max_score : satisfaction score 의 합 중 가장 큰 값을 저장하기 위한 변수

index : store 배열에 값을 차례로 저장할 때 인덱스의 위치를 가리키는 변수

```
Satisfaction List[200], store[21], result[21];  
int check_items[21], check_member[21];  
int max_score = 0, indexx = 1;
```

<함수 설명>

Read_File : satisfaction.txt 파일을 읽기전용으로 불러온 다음, 정수부분을 위에서 생성한 List 구조체 배열에 저장해줍니다. i 변수를 이용해서 member, item, score 을 차례차례 저장해주도록 하였습니다.

```
void Read_File(void)
{
    int i = 0;
    char s[100];
    FILE *fp = fopen("satisfaction.txt", "r");
    fgets(s, 100, fp);
    while (fscanf(fp, "%d %d %d\n", &List[i].member, &List[i].item, &List[i].score))
        i++;
    fclose(fp);
}
```

Init : 방문처리한 배열과 store 배열의 인덱스를 가리키는 변수를 초기값으로 초기화 시켜주는 함수입니다.

```
void Init(void)
{
    for (int i = 1 ; i <= LENGTH ; i++)
    {
        check_items[i] = 0;
        check_member[i] = 0;
    }
    indexx = 1;
}
```

Pick_random : 1 부터 20 중 아직 선택되지 않은 Member 를 check_member 배열을 이용하여 방문처리 되지 않은 member 중 랜덤으로 선택하는 Random 함수입니다

```
int Pick_random(void)
{
    srand(time(NULL));
    int ran = rand() % LENGTH + 1;
    while (check_member[ran])
        ran = rand() % LENGTH + 1;
    return ran;
}
```

Get_item : k 번째 Member 가 선택할 수 있는 item 중 가장 큰 score 을 가진 item 을 선택해주는 함수입니다. 반복문을 이용해서 List 구조체 배열에 저장된 member 의 값과 k 가 동일할 때까지 인덱스를 증가시켜주고, k 와 동일한 값을 가진 List 구조체 배열 인덱스 중 아직 선택되지 않은 가장 큰 score 을 가진 item 을 찾습니다. 그 후 store 구조체 배열의 indexx 번째 인덱스에 선택된 member, item, score 을 저장해주고, check_member, check_item 에 각각 선택된 인덱스의 값에 1 을 저장해줍니다. item 이 선택 되었는지 확인하기 위해 check_item 의 선택된 item 인덱스의 값을 반환해줍니다. 만약 item 이 선택되지 않았다면, 0 번째 check_item 인덱스의 값을 반환하는데, 이는 0 이므로 0 을 반환합니다.

```
int Get_item(int k)
{
    int maximum = 0;
    int select = 0;
    int i = 0;
    while (List[i].member != k)
        i++;
    while (List[i].member == k)
    {
        if (check_items[List[i].item] == 0 && maximum < List[i].score)
        {
            maximum = List[i].score;
            select = List[i].item;
        }
        i++;
    }
    if (maximum != 0)
    {
        store[indexx].member = k;
        store[indexx].item = select;
        store[indexx].score = maximum;
        check_member[k] = 1;
        check_items[select] = 1;
    }
    return check_items[select];
}
```

Is_Finish : 모든 item 이 전부 선택되었으면 1, 아니면 0 을 반환해주는 함수입니다.

```
int Is_Finish(void)
{
    for (int i = 1 ; i <= LENGTH ; i++)
    {
        if (check_items[i] == 0)
            return 0;
    }
    return 1;
}
```

Random_Selection : 위의 함수들을 이용하여 모든 member 들이 Item 을 모두 선택 할 때 까지 반복하고, 저장된 score 의 합을 반환해주는 함수입니다. Is_Finish()함수를 반복문 탈출조건으로 사용하여 모든 item 이 선택될 때 까지 Pick_random()으로 랜덤하게 member 를 선택한 다음, Get_item()의 매개변수로 넣어서 그 member 가 선택할 수 있는 최대의 score 을 가진 item 을 선택합니다. 하지만 item 이 선택되지 않았을 수 있으므로, 조건문에 item 이 선택되지 않았다면 init()을 이용해 초기화하고 새로 시작하도록 하고, 정상적으로 선택되었다면 Indexx 값을 증가시켜주며 store 의 다음 인덱스를 가리킬 수 있도록 하였습니다.

모든 member 에 대한 item 이 선택되었다면, 반복문을 이용해 모든 score 을 더해주고, 반환해줍니다.

```
int Random_Selection(void)
{
    int sum = 0;
    int temp = 0;
    while(!Is_Finish())
    {
        temp = Get_item(Pick_random());
        if (!temp)
            Init();
        else
            indexx++;
    }
    for (int i = 1 ; i <= LENGTH ; i++)
        sum = sum + store[i].score;
    Init();
    return sum;
}
```

Print : 콘솔 창에 각 Member 가 어떤 item 을 가져갔고, 이에 따른 Satisfaction Score 를 선택된 순서대로 출력하고, 마지막 줄에 Max_Score Sum 을 출력해 주는 함수입니다.

```
void Print(void)
{
    for (int i = 1 ; i <= LENGTH ; i++)
        printf("%2d Member's item : %2d, Score : %2d\n", result[i].member, result[i].item, result[i].score);
    printf("Total result : %d\n", max_score);
}
```

Main 문 : 임의로 정한 값인 n(반복 횟수), s(max_score의 한계값)를 반복문 탈출조건으로 주어서 Random_Selection()을 탈출할 때 까지 반복적으로 실행합니다. n이 증가할 수록 더욱 최대값에 가까운 값을 찾을 확률이 증가하고, s의 크기가 증가할수록 max_score의 한계값이 증가하기 때문에 적당히 큰 값을 특정합니다. 실행할 때마다 Random_Selection()의 값이 커지면 최신화 시켜주면서 최대값을 max_score에 저장하고, 그 때 저장된 store 구조체 배열의 값들을 result 구조체 배열 안에 저장합니다. 반복문이 끝난 후, 최대값을 가질 때의 result 구조체 배열의 값과 max_score을 Print()를 이용하여 출력합니다.

```
int main(void)
{
    int n = 1000, s = 1800, epoch = 0, temp = 0;
    Read_File();
    while (epoch < n && max_score < s)
    {
        temp = Random_Selection();
        epoch++;
        if (max_score < temp)
        {
            max_score = temp;
            for (int i = 1 ; i <= LENGTH ; i++)
            {
                result[i].member = store[i].member;
                result[i].item = store[i].item;
                result[i].score = store[i].score;
            }
        }
    }
    Print();
    return 0;
}
```

3. 결과물 출력

랜덤하게 member를 선택하므로 실행할 때마다 결과가 달라지는 것을 확인할 수 있다.

```
"C:\Users\heath\OneDrive\Documents\codeblock source\Project\project.exe"
14 Member's item : 7, Score : 95
7 Member's item : 2, Score : 83
16 Member's item : 6, Score : 94
5 Member's item : 15, Score : 86
15 Member's item : 20, Score : 89
6 Member's item : 16, Score : 92
17 Member's item : 12, Score : 93
9 Member's item : 11, Score : 80
3 Member's item : 3, Score : 80
1 Member's item : 18, Score : 93
2 Member's item : 8, Score : 91
18 Member's item : 19, Score : 99
13 Member's item : 1, Score : 94
12 Member's item : 13, Score : 79
4 Member's item : 9, Score : 55
10 Member's item : 4, Score : 74
20 Member's item : 17, Score : 74
19 Member's item : 10, Score : 76
8 Member's item : 5, Score : 56
11 Member's item : 14, Score : 70
Total result : 1653

Process returned 0 (0x0)   execution time : 0.097 s
Press any key to continue.
```

```
"C:\Users\heath\OneDrive\Documents\codeblock source\Project\project.exe"
15 Member's item : 20, Score : 89
3 Member's item : 12, Score : 80
11 Member's item : 16, Score : 77
16 Member's item : 6, Score : 94
19 Member's item : 4, Score : 96
18 Member's item : 19, Score : 99
1 Member's item : 18, Score : 93
5 Member's item : 15, Score : 86
14 Member's item : 7, Score : 95
7 Member's item : 2, Score : 83
9 Member's item : 11, Score : 80
10 Member's item : 13, Score : 95
12 Member's item : 1, Score : 84
8 Member's item : 3, Score : 80
17 Member's item : 17, Score : 63
6 Member's item : 10, Score : 54
4 Member's item : 8, Score : 84
2 Member's item : 5, Score : 31
13 Member's item : 9, Score : 54
20 Member's item : 14, Score : 70
Total result : 1587

Process returned 0 (0x0)   execution time : 5.019 s
Press any key to continue.
```