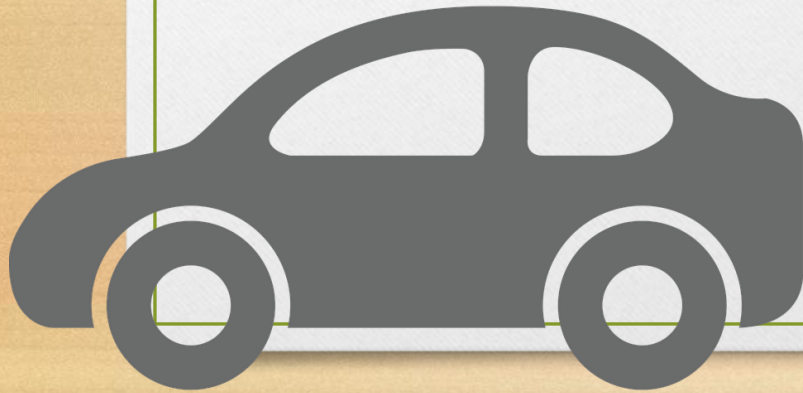


주차장 관리 프로그램



학부 : Ai융합학부

학번 : 20213062

이름 : 김성윤

목차

- 1) 프로그램 소개
- 2) 기능소개
- 3) 구현방법
- 4) 발전방향

1) 프로그램 소개



차량 정보에
주차장 위치 저장

2)기능소개 (with 결과 화면)

2-1) '주차' 기능

2-2) '출차' 기능

2-3) '차량 위치 검색' 기능

2-4) '현 주차장 상황' 기능

2-5) '종료' 기능

2-6) '파일 입출력' 기능

```
berry6@ejlee-lecture:~/For-school-programing$ ./proje
```

```
=====<MENU>=====
```

1. 주 차
2. 출 차
3. 차 량 위 치 검 색
4. 현 주 차 상 황
0. 종 료

```
번호 선택 =>1
```

```
번호 판 들 입 력 하 세 요 .:1234567
```

```
=====<1>=====
```

```
0 0 X 0 0 0 0 0 X 0 X 0
```

```
=====<2>=====
```

```
X 0 0 0 X 0 X 0 0 0 0 X
```

```
=====<3>=====
```

```
0 0 X 0 X X 0 0 0 X 0 X
```

```
주 차 할 위 치 를 선택 하 세 요 .: (층 , n번 패 ) 1 1
```

```
=====<1>=====
```

```
X 0 X 0 0 0 0 0 X 0 X 0
```

```
=====<2>=====
```

```
X 0 0 0 X 0 X 0 0 0 0 X
```

```
=====<3>=====
```

```
0 0 X 0 X X 0 0 0 X 0 X
```

```
=====<MENU>=====
```

1. 주 차
2. 출 차
3. 차 량 위 치 검 색
4. 현 주 차 상 황
0. 종 료

```
번호 선택 =>
```

2)기능소개 (with 결과 화면)

2-1) '주차' 기능

2-2) '출차' 기능

2-3) '차량 위치 검색' 기능

2-4) '현 주차장 상황' 기능

2-5) '종료' 기능

2-6) '파일 입출력' 기능

```
=====<MENU>=====
1. 주 차
2. 출 차
3. 차 량 위 치 검 색
4. 현 주 차 상 황
0. 종 료
번호 선택 =>2
차 량 의 번 호 판 을 입 력 하 세 요 :1234567
삭 제 가 완 료 되 었 습 니 다 .
```

```
=====<MENU>=====
1. 주 차
2. 출 차
3. 차 량 위 치 검 색
4. 현 주 차 상 황
0. 종 료
번호 선택 =>4
=====<1>=====
O O X O O O O O X O X O
=====<2>=====
X O O O X O X O O O O X
=====<3>=====
O O X O X X O O O X O X
=====<MENU>=====
1. 주 차
2. 출 차
3. 차 량 위 치 검 색
4. 현 주 차 상 황
0. 종 료
번호 선택 =>
```


2)기능소개 (with 결과 화면)

2-1) '주차' 기능

2-2) '출차' 기능

2-3) '차량 위치 검색' 기능

2-4) '현 주차장 상황' 기능

2-5) '종료' 기능

2-6) '파일 입출력' 기능

```
=====<MENU>=====
1. 주차
2. 출차
3. 차량 위치 검색
4. 현 주차 상황
0. 종료
번호선택 =>1
번호판을 입력하세요.:1234567

=====<1>=====
O O X O O O O O X O X O
=====<2>=====
X O O O X O X O O O O X
=====<3>=====
O O X O X X O O O X O X
주차할 위치를 선택하세요.:(층, n번째)1 1
=====<1>=====
X O X O O O O O X O X O
=====<2>=====
X O O O X O X O O O O X
=====<3>=====
O O X O X X O O O X O X
=====<MENU>=====
1. 주차
2. 출차
3. 차량 위치 검색
4. 현 주차 상황
0. 종료
번호선택 =>3
차량의 번호판을 입력하세요.:1234567
1층, 1번째
=====<MENU>=====
1. 주차
2. 출차
3. 차량 위치 검색
4. 현 주차 상황
0. 종료
번호선택 =>
```

2)기능소개 (with 결과 화면)

2-1) '주차' 기능

2-2) '출차' 기능

2-3) '차량 위치 검색' 기능

2-4) '현 주차장 상황' 기능

2-5) '종료' 기능

2-6) '파일 입출력' 기능

```
=====<MENU>=====
1. 주 차
2. 출 차
3. 차 량 위 치 검 색
4. 현 주 차 상 황
0. 종 료
번호 선택 =>4
=====<1>=====
X O X O O O O O X O X O
=====<2>=====
X O O O X O X O O O O X
=====<3>=====
O O X O X X O O O X O X
=====<MENU>=====
1. 주 차
2. 출 차
3. 차 량 위 치 검 색
4. 현 주 차 상 황
0. 종 료
번호 선택 =>
```

2)기능소개 (with 결과 화면)

2-1) '주차' 기능

2-2) '출차' 기능

2-3) '차량 위치 검색' 기능

2-4) '현 주차장 상황' 기능

2-5) '종료' 기능

2-6) '파일 입출력' 기능

```
=====<MENU>=====
```

```
1. 주 차
2. 출 차
3. 차 량 위 치 검 색
4. 현 주 차 상 황
0. 종 료
```

```
번호 선택 =>4
```

```
=====<1>=====
```

```
X O X O O O O X O X O
```

```
=====<2>=====
```

```
X O O O X O X O O O O X
```

```
=====<3>=====
```

```
O O X O X X O O O X O X
```

```
=====<MENU>=====
```

```
1. 주 차
2. 출 차
3. 차 량 위 치 검 색
4. 현 주 차 상 황
0. 종 료
```

```
번호 선택 =>0
```

```
berry6@ejlee-lecture:~/For-school-programing$
```


2) 기능 소개 (with 결과 화면)

2-1) '주차' 기능

2-2) '출차' 기능

2-3) '차량 위치 검색' 기능

2-4) '현 주차장 상황' 기능

2-5) '종료' 기능

2-6) '파일 입출력' 기능

```
berry6@ejlee-lecture:~/For-school-programing$ vi input_data.txt
```

```
1 11111111 2 1
2 12121233 3 3
3 12312512 3 6
4 33223111 3 5
5 34593929 1 3
6 14241234 1 9
7 12312414 1 11
8 14123332 2 7
9 12412412 2 12
10 12412125 3 12
11 12312421 3 10
12 11241244 2 5
```

마지막으로
입력한
1234567로
output 증가

```
berry6@ejlee-lecture:~/For-school-programing$ vi output_data.txt
```

```
1 My first data file
2 Car_number : À@zÈp^?, Car_location : 1i,µlë<88>i$.
3 Car_number : À@zÈp^?, Car_location : 2i,µ5ë<88>i$.
4 Car_number : À@zÈp^?, Car_location : 3i,µ10ë<88>i$.
5 Car_number : À@zÈp^?, Car_location : 3i,µ12ë<88>i$.
6 Car_number : À@zÈp^?, Car_location : 2i,µ12ë<88>i$.
7 Car_number : À@zÈp^?, Car_location : 2i,µ7ë<88>i$.
8 Car_number : À@zÈp^?, Car_location : 1i,µ11ë<88>i$.
9 Car_number : À@zÈp^?, Car_location : 1i,µ9ë<88>i$.
10 Car_number : À@zÈp^?, Car_location : 1i,µ3ë<88>i$.
11 Car_number : À@zÈp^?, Car_location : 3i,µ5ë<88>i$.
12 Car_number : À@zÈp^?, Car_location : 3i,µ6ë<88>i$.
13 Car_number : À@zÈp^?, Car_location : 3i,µ3ë<88>i$.
14 Car_number : À@zÈp^?, Car_location : 2i,µ1ë<88>i$.
```

3) 구현 방법

```
1 #include<string.h>
2 #include<stdio.h>
3 #include<stdlib.h>
4 #define SIZE 12
5 #define FLOOR 3
6 #define numsize 8
7
8 struct node{
9     char num[numsize];
10    int floor;
11    int X;
12    struct node* next;
13 };
14
15 int space[FLOOR][SIZE];
16
17 void menu()
18 {
19     printf("=====<MENU>=====\\n");
20     printf("1. 주 자 \\n");
21     printf("2. 출 자 \\n");
22     printf("3. 자 랑 위 치 검 색 \\n");
23     printf("4. 전 주 자 상 황 \\n");
24     printf("0. 종 료 \\n");
25     printf("번 호 선택 =>");
26 }
27
28 void ResetSpace()
29 {
30     int k,i;
31     for (i=0;i<FLOOR;i++){
32         for (k=0;k<SIZE;k++){
33             space[i][k]='O';
34         }
35     }
36 }
37
38 void spaceNow()
39 {
40     int a,b;
41     for(a=0;a<FLOOR;a++){
42         printf("=====<sd>=====\\n", a+1);
43         for (b=0;b<SIZE;b++){
44             printf(" %c ", space[a][b]);
45         }
46         printf("\\n");
47     }
48 }
49 void change_space(int tfloor, int tX)
50 {
51     while(1){
52         if (space[tfloor-1][tX-1]=='O'){
53             space[tfloor-1][tX-1]='X';
54             break;}
55
56         if (space[tfloor-1][tX-1]=='X'){
57             space[tfloor-1][tX-1]='O';
58             break;}
59     }
60 }
```

3) 구현 방법

```
61 struct node insCar_for_data(struct node** list_head, int get_floor, int get_X, char name[numsize], void (*func)(char*, char*))
62 {
63     struct node* new_node;
64     if (space[get_floor-1][get_X-1]=='O'){
65         new_node =
66             (struct node*) malloc (sizeof(struct node));
67         func(new_node->num, name);
68         new_node->floor = get_floor;
69         new_node->X = get_X;
70         (new_node->next) = *list_head;
71         *list_head = new_node;
72         change_space(get_floor, get_X);}
73     else
74         printf("이미 존재하는 자리입니다.\n");
75 }
76
77 struct node insCar(struct node** list_head, void (*func)(char*, char*))
78 {
79     int serv_floor, serv_X;
80     struct node* new_node;
81     char serv_num[numsize];
82     printf("번호판을 입력하세요. :");
83     scanf("%s", serv_num);
84     spacenow();
85     printf("주차장 위치를 선택하세요.:(층, 번째)");
86     scanf("%d %d", &serv_floor, &serv_X);
87     if (space[serv_floor-1][serv_X-1]=='O'){
88
89         new_node =
90             (struct node*) malloc (sizeof(struct node));
91         func(new_node->num, serv_num);
92         new_node->floor = serv_floor;
93         new_node->X = serv_X;
94         (new_node->next) = *list_head;
95         *list_head = new_node;
96         change_space(serv_floor, serv_X);}
97     else
98         printf("잘못 입력하셨습니다.\n");
99     spacenow();
100 }
101
102 void insnum(char a[],char b[])
103 {
104     for(int i=0;i<numsize;i++){a[i] = b[i];}
105 }
106
107 struct node delCar(struct node* tmp, struct node** list_head, void (*func)(char*, char*))
108 {
109     int result;
110     char get_num[numsize];
111     struct node* new_node;
112     struct node* jump;
113     struct node* prv_node=NULL;
114     printf("차량의 번호판을 입력하세요. :");
115     scanf("%s", get_num);
116     fflush(stdin);
117
118     new_node =
119         (struct node*) malloc (sizeof(struct node));
120     func(new_node->num, get_num);
```


3) 구현 방법

```
121 tmp = *list_head;
122
123 while(tmp){
124     result=strcmp(tmp->num, new_node->num);
125     if(result==0){
126         if((tmp->next==NULL)&&(tmp == *list_head)){
127             *list_head = tmp->next;
128             printf("식재가 완료되었습니다.\n");
129             change_space(tmp->floor,tmp->X);
130             break;}
131         else if(tmp == *list_head){
132             *list_head = tmp->next;
133             printf("식재가 완료되었습니다.\n");
134             change_space(tmp->floor,tmp->X);
135             break;}
136         jump = tmp->next;
137         prv_node->next = jump;
138         printf("식재가 완료되었습니다.\n");
139         change_space(tmp->floor,tmp->X);
140         break;}
141     prv_node = tmp;
142     tmp = tmp->next;
143
144 }
145 }
146
147 struct node findCar(struct node* tmp, struct node** list_head, void (*func)(char*, char*))
148 {
149     int find_floor, find_X;
150     struct node* new_node;
151     char get_num[numsize];
152     int result;
153     tmp = *list_head;
154     if(tmp==NULL){printf("주차장에 차량이 없습니다.\n");}
155     else{
156         printf("차량의 번호판을 입력하세요:");
157         scanf("%s", get_num);
158
159         new_node =
160             (struct node*) malloc (sizeof(struct node));
161         func(new_node->num, get_num);}
162
163     while(tmp){
164         result = strcmp(tmp->num, new_node->num);
165         if(result==0){
166             find_floor = tmp->floor;
167             find_X = tmp->X;
168             printf("%d층, %d번째\n", find_floor, find_X);
169             break;}
170         if(tmp->next==NULL){
171             printf("검색결과가 없습니다.\n");
172             break;}
173         tmp = tmp->next;
174     }
175 }
176
177 struct node File_w(struct node* tmp, struct node** list_head, void (*func)(char*, char*))
178 {
179     int data_floor, data_X;
180     char data_num[numsize];
```

3) 구현 방법

```
181 FILE *fp = NULL;
182 fp = fopen("output_data.txt", "w");
183
184 if(fp == NULL){
185     printf("Failed to open file %s\n", "output_data.txt");
186 }
187 fprintf(fp, "My first data file\n");
188 tmp = *list_head;
189 while(tmp){
190     data_floor = tmp->floor;
191     data_X = tmp->X;
192     func(tmp->num, data_num);
193     fprintf(fp, "Car_number : %s, Car_location : %d을 %d번 째 \n", data_num, data_floor, data_X);
194     tmp = tmp->next;}
195 fclose(fp);
196 }
197
198
199 struct node* File_r(struct node** list_head)
200 {
201     FILE *fp = NULL;
202     char buff[numsize];
203     int floor, X;
204     fp = fopen("input_data.txt", "r");
205
206     if(fp == NULL){
207         printf("Failed to open file %s\n", "input_data.txt");
208     }
209     while(fscanf(fp, "%s %d %d", buff, &floor, &X) != EOF){
210         insCar_for_data(list_head, floor, X, buff, insnum);
211     }
212     fclose(fp);
213     return 0;
214 }
215
216
217 int main()
218 {
219     struct node* tmp;
220     struct node* list_head=NULL;
221     int answer;
222     int check=1;
223     Resetspace();
224     File_r(&list_head);
225     while(check!=0)
226     {
227         menu();
228         scanf("%d", &answer);
229         fflush(stdin);
230
231
232         switch(answer){
233             case 1: insCar(&list_head, insnum);continue;
234             case 2: delCar(tmp, &list_head, insnum);continue;
235             case 3: findCar(tmp, &list_head, insnum);continue;
236             case 4: spacenow();continue;
237             case 0: check=0;break;
238             default :
239                 printf("잘못 입력 하셨습니다.\n");
240         }
```

3) 구현 방법

```
241 }
242
243 File_w(tmp, &list_head, insnum);
244 return 0;
245 }
246
247
```

```
181 FILE *fp = NULL;
182 fp = fopen("output_data.txt", "w");
183
184 if(fp == NULL){
185     printf("Failed to open file %s\n", "output_data.txt");
186 }
187
188 fprintf(fp, "My first data file\n");
189 tmp = *list_head;
190 while(tmp){
191     data_floor = tmp->floor;
192     data_X = tmp->X;
193     func(tmp->num, data_num);
194     fprintf(fp, "Car_number : %s, Car_location : %d을 %d번 째 \n", data_num, data_floor, data_X);
195     tmp = tmp->next;}
196 fclose(fp);
197 }
198
199 struct node* File_r(struct node** list_head)
200 {
201     FILE *fp = NULL;
202     char buff[numsize];
203     int floor, X;
204     fp = fopen("input_data.txt", "r");
205
206     if(fp == NULL){
207         printf("Failed to open file %s\n", "input_data.txt");
208     }
209
210     while(fscanf(fp, "%s %d %d", buff, &floor, &X) != EOF){
211         insCar_for_data(list_head, floor, X, buff, insnum);
212     }
213     fclose(fp);
214     return 0;
215 }
216
217 int main()
218 {
219     struct node* tmp;
220     struct node* list_head=NULL;
221     int answer;
222     int check=1;
223     Resetspace();
224     File_r(&list_head);
225     while(check!=0)
226     {
227         menu();
228         scanf("%d", &answer);;
229         fflush(stdin);
230
231
232         switch(answer){
233             case 1: insCar(&list_head, insnum);continue;
234             case 2: delCar(tmp, &list_head, insnum);continue;
235             case 3: findCar(tmp, &list_head, insnum);continue;
236             case 4: spacenow();continue;
237             case 0: check=0;break;
238             default :
239                 printf("잘못 입력 하셨습니다.\n");
240         }
241     }
242 }
```


3) 구현 방법

```
1 11111111 2 1
2 12121233 3 3
3 12312512 3 6
4 33223111 3 5
5 34593929 1 3
6 14241234 1 9
7 12312414 1 11
8 14123332 2 7
9 12412412 2 12
10 12412125 3 12
11 12312421 3 10
12 11241244 2 5
```

3) 구현 방법

- 총 246줄 c언어 파일과 input_data.txt 파일을 통해서 구현할 수 있다.
- (추가적으로 갈아 넣은 수면시간과 당겨쓴 생명줄도...)

5) 발전 방향

- 한 사용자에게 여러 대의 자동차 정보 저장
- 간략한 코드
- 더 많은 흐름 설명

(예, '잘못된 값을 입력하셨습니다.', '리스트에 없는 정보입니다. oo기능이 종료됩니다.'))

FINISH

