

The screenshot shows the assembly code for a function named 'main'. The code is written in C-like pseudo-assembly. It prompts the user for input, reads it into a buffer 's', removes the newline character, and then prints a message indicating that the password is incorrect. The assembly code is as follows:

```
1 int64 __fastcall main(int a1, char **a2, char **a3)
2 {
3     printf("Enter the password: ");
4     fflush(stdout);
5     fgets(s, 256, stdin);
6     s[strcspn(s, "\n")] = 0;
7     sub_5540(s);
8     puts("That's not the password...");
9     return 1LL;
0 }
```

전역 변수 s에 사용자 입력을 받아 함수 인자로 넘긴다.

```
9
0     if ( *a1 == 106 )
1     {
2         v36 = a1 + 1;
3         if ( *v36 == 117 )
4         {
5             v35 = v36 + 1;
6             if ( *v35 == 109 )
7             {
8                 v34 = v35 + 1;
9                 if ( *v34 == 112 )
0                 {
1                     v33 = v34 + 1;
2                     if ( *v33 == 105 )
3                     {
4                         v32 = v33 + 1;
5                         if ( *v32 == 110 )
6                         {
7                             v31 = v32 + 1;
8                             if ( *v31 == 103 )
9                             {
0                                 v30 = v31 + 1;
```

해당 함수에서는 수많은 조건문을 통해 사용자가 입력한 문자를 한글자씩 비교한다.

```
    v2 = v3 + 1;
    if ( *v2 == 110 )
    {
        v1 = v2 + 1;
        if ( *v1 == 101 )
            sub_1200(v1 + 1);
    }
}
```

모든 조건문을 만족하면 함수의 인자로 사용자의 문자열이 들여가고, 함수 내부에서 FLAG를 출력해준다.

```
1 void __fastcall __noretturn sub_1200(_BYTE *a1)
2 {
3     if ( !*a1 )
4     {
5         printf("The flag is actf{%s}\n", s);
6         exit(0);
7     }
8     puts("That's not the password...");
9     exit(1);
0 }
```

```
Enter the password: jumping_my_way_to_the_flag_one_by_one
The flag is actf{jumping_my_way_to_the_flag_one_by_one}
```