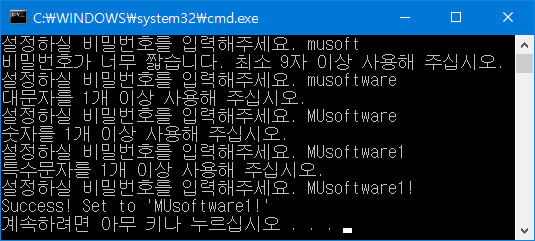
|  |
| --- |
| 비밀번호로 적합한지를 검사하는 필터 |
| ◆ 소스 코드 |
| #include <stdio.h>  #include <stdlib.h>  #include <string.h>  #include <ctype.h>  //가능한 특수 문자인지를 판단해주는 함수  int isschar(char input\_char) {  if (input\_char == 33) return 1;//!  if (input\_char == 35) return 1;//#  if (input\_char == 36) return 1;//$  if (input\_char == 63) return 1;//?  if (input\_char == 64) return 1;//@  return 0;  }  int main(void) {  char input[100] = { 0 }, password[100] = { 0 };  while (1) {  int is\_str\_ok = 0, count = 0;  //암호 문자열을 저장할 공간을 NULL로 초기화해주고,  for (int n = 0; n < 100; n++) input[n] = 0;  printf("설정하실 비밀번호를 입력해주세요. ");  gets\_s(input);//입력을 받습니다.  //연산  for (count; input[count]; count++) { //여기서 비밀번호의 글자 개수도 세줍니다.  /\*최소한 대문자, 소문자, 숫자, 특정한 특수문자(!, @, #, $, ?) 1개씩은 들어가도록.  \*1, 2, 4, 8은 0b0001, 0b0010, 0b0100, 0b1000으로,  \*활성화된 이진수로 뭐가 참이고 거짓인지 알 수 있습니다.  \*뭐, 꼼수긴 하지만요;  \*배열을 써도 괜찮긴 한데 메모리 절약을 위해서!(...)  \*/  if (isupper(input[count]) && !(is\_str\_ok & 1)) is\_str\_ok += 1;//대문자인지?  else if (islower(input[count]) && !(is\_str\_ok & 2)) is\_str\_ok += 2;//대문자가 아니면 소문자?  else if (isdigit(input[count]) && !(is\_str\_ok & 4)) is\_str\_ok += 4;//소문자도 아니면 숫자?  else if (isschar(input[count]) && !(is\_str\_ok & 8)) is\_str\_ok += 8;//저것전부 아니면 허용된 특수문자?  }  if (count < 9) { printf("비밀번호가 너무 짧습니다. 최소 9자 이상 사용해 주십시오.\n"); continue; }//9글자 이상인지?  else if (is\_str\_ok != 15) {//어라, 필요한 글자가 없다?  if (!(is\_str\_ok & 1)) { printf("대문자를 1개 이상 사용해 주십시오.\n"); continue; }  if (!(is\_str\_ok & 2)) { printf("소문자를 1개 이상 사용해 주십시오.\n"); continue; }  if (!(is\_str\_ok & 4)) { printf("숫자를 1개 이상 사용해 주십시오.\n"); continue; }  if (!(is\_str\_ok & 8)) { printf("특수문자를 1개 이상 사용해 주십시오.\n"); continue; }  }  else break;//모든 조건에 맞으므로 While 탈출!  }  //이제 입력 받은 문자열을 저장하고...  //(물론 실전에서 이렇게 하면 안되고 SHA-3같은걸로 암호화 해줘야 합니다.)  strcpy(password, input);  printf("Success! Set to '%s'\n", password);//잘 저장되었다고 출력해줍니다.  return 0;//0 반환하고 종료!  } |

◆ 설명

주석을 참조해주세요 :-)

◆ 실행 화면



모든 코드는 https://github.com/MU-Software/hs\_c/tree/master/1108에서 확인하실 수 있습니다.