プログラミング１ および演習　 レポートファイル

学籍番号： v24121　 氏名：樋口　陽輝

課題 p7-1.c

確保したメモリ領域を超えて配列に値を格納すると、別の配列で確保できている領域に格納し、書き換える場合があることが分かりました。

課題 p7-2.c

for 文を用いて配列に代入された値を順番に参照し、配列内の値を全て合算させる方法が分かりました。

課題 p7-3.c

¥0 が特殊文字であり、出力の際はあってもなくても視覚的には同じ動作をすることが分かりました。また、¥0 が配列内に存在した場合、後ろに文字があってもそこで出力が終了することが分かりました。

課題 p7-4.c

文字列の最後の¥0 を参照することで、id[]が 0 を返し while(0)となり、処理が終了することがわかりました。また、1 文字ずつ取り出すため、上書きが簡単に出来ることが分かりました。

課題 p7-5.c

継続条件を(str[i]!='¥0')にして、i++を繰り返すことで、文字数を取り出すことが出来ると分かりました。また、一文字ずつ参照しながら代入することで、文字列のコピーが出来ることが分かりました。

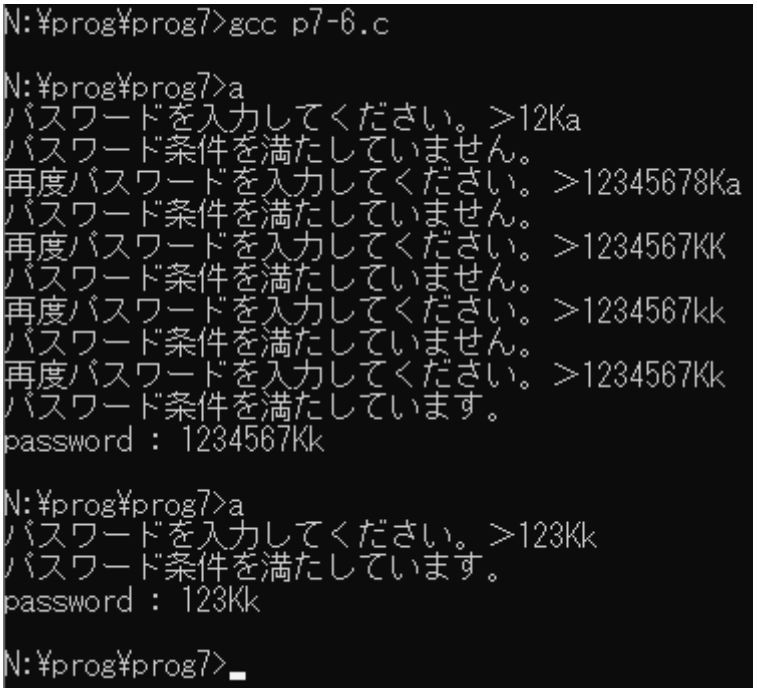
注意）こちらのページの内容にソースコードや結果のキャプチャ画面は要りません。

問題作成課題（p7-6.c）

プログラムソースコード画像
( プログラムが長くなるが小さくしすぎないこと。 )

```
/* p7-6.c */

#include<stdio.h>
#include<string.h>

int main(){
        int flagl=0, flags=0, flagb=0, flagn=0, i,j=0;
        char pass[20];

        printf("パスワードを入力してください。>");
        gets(pass);
        while(pass[j] != '¥0'){
                j++;
        }

        for( i=0; i<=j; i++){
                if('a'<= pass[i] && pass[i] <='z'){
                        flags = 1;
                }

                if('A'<= pass[i] && pass[i] <='Z'){
                        flagb = 1;
                }

                if('0'<= pass[i] && pass[i] <='9'){
                        flagn = 1;
                }

                if(j>=5 && j<10){
                        flagl = 1;
                }

        }

        if(flagb == 1 && flags == 1 && flagn == 1 && flagl == 1){
                printf("パスワード条件を満たしています。¥n");
                printf("password : %s¥n",pass);
        }
        else{
                printf("パスワード条件を満たしていません。¥n");
                printf("再度");
                main();
        }

        return 0;
}
/*
配列に文字列を入力して、格納された文字を参照し、
条件を満たすことを確認するプログラムの記述方法が分かりました。
*/
```

実行結果画像




本日の感想や反省

文字列を用いて、配列の扱い方を知ることが出来ました。最後は記述量が多いこともあり、なかなか期待通りの動作をせず苦労しましたが、要件通りのプログラムを記述することが出来ました。