

[1] 課題内容

入力した正の整数を降順に並べ換えて出力するプログラムを作成せよ。
入力データは50以下とし、以下の数が混在しているとする。

16進数：先頭1文字がxまたはX(エックスの小文字か大文字)

8進数：先頭1文字が0 (零)

10進数：先頭1文字が0 (零) 以外の数字

考察ポイント：複数あるポインタ変数が何を指しているかの把握

Makeの理解

コンパイルから実行までの流れ

複数言語プログラムの結合と実行

[2] それぞれの関数プログラムとmakefileファイル

a. asmプログラム

```
-----
1  /*****
2  Program   : asm.c
3  Student-ID : 045713C
4  Author    : OSHIRO, Kazuya
5  Date      : 04/07/19
6  Comment   : 基数変換と整列処理
7  *****/
8
9  #include <stdio.h>
10 #include <string.h>
11 #define MAX 256 /* MAXを256に定義する */
12
13 void conv10 (char **x, int *k, int n);
14 void select_sort (int x[], int m[], int n);
15 void print_num (char *x[], int m[], int n);
16 void msg(); /* それぞれの関数の宣言 */
17
18 main()
19 {
20     char *dt[50], num[10], buf[MAX], *p=buf;
21     /* char型の配列、ポインタ、ポインタ配列を宣言 */
22     int n=0, len, i10[50], move[50]; /* 整数型の変数、配列の宣言 */
23
24     puts("----- Input"); /* 文字の出力 */
25     while(gets(num) != NULL){ /* 標準入力の1列がNULLになるまで繰り返す */
26         len = strlen(num);
27         /* 変数lenに文字列numの長さを代入(NULL文字は含まない) */
28         if (p > buf+MAX-(len+1)) break;
29         /* ポインタpがbufの範囲外を指したときwhile文を終了する */
30         strcpy (p, num); /* 文字列numをpにコピーする */
31         dt[n] = p; /* dtにpが指すアドレスを代入 */

```

```

32     p += len + 1;          /* pのアドレスをその一列分ずらす */
33     n++;                  /* nをインクリメントする */
34 }
35 puts("----- Result"); /* 文字を出力 */
36 conv10(dt, i10, n);      /* conv10関数を呼び出す */
37 select_sort(i10, move, n); /* select_sort関数を呼び出す */
38 print_num(dt, move, n); /* print_num関数を呼び出す */
39 msg();                  /* msg関数を呼び出す */
40 }

```

b. conv10プログラム

```

1  /*****
2  Program   : conv10.c
3  Comment   : 基数判断と変換
4  *****/
5
6  void conv10(char **x, int *k, int n) /* conv10を定義する */
7  {
8      while (n-- > 0){          /* nが0になるまで以下のことを繰り返す */
9          switch (**x){
10             case '0':         /* **xが0のとき */
11                 sscanf(*x+1, "%o", k);
12                 /* *xに1を足しポインタを移動しその後の文字列を八進数書式に変換しkに代入 */
13                 break;        /* switch文を抜ける */
14             case 'x':
15             case 'X':         /* **xがxもしくはXのとき */
16                 sscanf(*x+1, "%x", k);
17                 /* *xに1を足しポインタを移動しその後の文字列を16進数書式に変換しkに代入 */
18                 break;        /* switch文を抜ける */
19             default:         /* 上記のcaseに当てはまらないとき*/
20                 sscanf(*x, "%u", k);
21                 /* *xに文字列を符号なしの整数として変換、後kに代入 */
22                 break;        /* switch文を抜ける */
23             }
24             k++;              /* kをインクリメントしてポインタの位置をずらす */
25             x++;              /* xをインクリメントしてdt[]の次の要素のアドレスを指す */
26         }
27     }

```

c. select_sortプログラム

```

1  /*****
2  Program   : select_sort.c
3  Comment   : 降順に並び替え (ソート)
4  *****/
5
6  void select_sort (int x[], int m[], int n) /* 関数定義 */
7  {
8      int i, j, k, w;          /* 整数型の変数の宣言 */
9
10     for(i=0; i<n; i++)
11         m[i] = i;
12     for(i=0; i<n-1; i++){

```

```

13     k = i;
14     for(j=i+1; j<n; j++)
15         if(x[k] < x[j])      /* kはx[i]~x[j]のなかで最大の要素の添字となる */
16             k = j;
17     w = x[i];                /* wにx[i]を代入する */
18     x[i] = x[k];            /* x[i]にx[k]を代入する */
19     x[k] = w;               /* x[k]にwを代入、それによりx[i]とx[k]を入れ替える */
20     w = m[i];               /* 上と同じ作業をmの配列にも行う */
21     m[i] = m[k];
22     m[k] = w;
23 }
24 }

```

d. print_numプログラム

```

1  /*****
2  Program   : print_num.c
3  Comment   : 文字列の出力
4  *****/
5
6  void print_num (char *x[], int m[], int n) /* print_num関数の定義 */
7  {
8      int i;
9      for (i=0; i<n; i++){
10         puts(x[m[i]]);
11         /* iを0に初期化、iがnより小さいのなら出力を行いその後、iをインクリメントする */
12     }
13 }

```

e. msgプログラム

```

1  *****/
2  Program   : msg.c
3  Comment   : メッセージ出力
4  *****/
5
6  void msg() /* msg関数の定義 */
7  {
8      printf("### MESSAGE FROM C ###by OSHIRO¥n"); /* 文字列を出力 */
9  }

```

f. makeファイル

```

#MakeFile
asm: asm.o conv10.o select_sort.o print_num.o msg.o
    cc asm.o conv10.o select_sort.o print_num.o msg.c -o asm
asm.o: asm.c
    cc -c asm.c
conv10.o: conv10.c
    cc -c conv10.c
select_sort.o: select_sort.c
    cc -c select_sort.c
print_num.o: print_num.c
    cc -c print_num.c

```

```
msg.o: msg.c
cc -c msg.c
```

[3] 出力結果

オブジェクトファイル生成

```
+++++
[nw0413:~/prog1/report/#8] j04013% ls -l
total 24
-rw-r--r--  1 j04013  j04013      1414 Jul 21 16:44 asm.c
-rw-r--r--  1 j04013  j04013      668 Jul 20 21:12 conv10.c
-rw-r--r--  1 j04013  j04013      299 Jul 21 12:47 makef
-rw-r--r--  1 j04013  j04013      232 Jul 20 12:40 msg.c
-rw-r--r--  1 j04013  j04013      376 Jul 21 12:59 print_num.c
-rw-r--r--  1 j04013  j04013      637 Jul 21 15:27 select_sort.c
```

```
[nw0413:~/prog1/report/#8] j04013% make -f makef
cc -c asm.c
cc -c conv10.c
cc -c select_sort.c
cc -c print_num.c
cc -c msg.c
cc asm.o conv10.o select_sort.o print_num.o msg.c -o asm
```

```
[nw0413:~/prog1/report/#8] j04013% ls -l
total 60
-rwxr-xr-x  1 j04013  j04013    15868 Jul 21 19:50 asm*
-rw-r--r--  1 j04013  j04013     1414 Jul 21 16:44 asm.c
-rw-r--r--  1 j04013  j04013     1908 Jul 21 19:50 asm.o
-rw-r--r--  1 j04013  j04013      668 Jul 20 21:12 conv10.c
-rw-r--r--  1 j04013  j04013     1116 Jul 21 19:50 conv10.o
-rw-r--r--  1 j04013  j04013      299 Jul 21 12:47 makef
-rw-r--r--  1 j04013  j04013      232 Jul 20 12:40 msg.c
-rw-r--r--  1 j04013  j04013      816 Jul 21 19:50 msg.o
-rw-r--r--  1 j04013  j04013      376 Jul 21 12:59 print_num.c
-rw-r--r--  1 j04013  j04013      768 Jul 21 19:50 print_num.o
-rw-r--r--  1 j04013  j04013      637 Jul 21 15:27 select_sort.c
-rw-r--r--  1 j04013  j04013      792 Jul 21 19:50 select_sort.o
```

プログラム実行結果

```
+++++
[nw0413:~/prog1/report/#8] j04013% ./asm
----- Input
warning: this program uses gets(), which is unsafe.
x21
021
055
22
----- Result
055
x21
22
021
### MESSAGE FROM C ###by OSHIRO
```

[4] 考察

asmプログラム

conv10プログラム

select_sortプログラム

print_numプログラム

*****switchについて*****

switch文はif/else文をいくつも並べたようなもので一般的な形式は次のようになる。

```
switch (式){
    case 定数1:
        文1
        . . .
        break;
    case 定数2:
        文2
        . . .
        break;
    .
    .
    .
    default:
        文
        . . .
        break;
}
```

switchの()の式と、caseのそれぞれの定数が比較され、一致するcaseに書かれている文を実行し、breakによって終了する。このとき一致するcaseがないときdefaultが実行されることになる。

*****makeについて*****

makeコマンドとは、モジュールが変更されたファイルを調べてmakefileなどのファイルから必要なソース・プログラムを自動的に判別し再コンパイルを行う。これにより、一部を変更した場合、全部のファイルを再コンパイルしなくても変更のあったファイルだけコンパイルし直しリンクすることができる。これより作業を効率的にでき、さらに局所的な見通しをよくしてバグを少なくする。といった利点があります。

ファイルの説明

ファイル名には標準的にMakeFile,MAKEFILEのような名前にする。このファイルにはプログラムを構築する方法を決定するためにmakeが使用するルールのようなものが納められている。

#がついたときコメント行を表す。

```
target: source [source2] [source3]
        comand
        [comand2]
        [comand3]
        . . .
```

targetは生成されるファイルの名前、ソースファイルからこのファイルが作られる。targetの生成に複数のソースファイルが必要なときそのソースファイルすべてを並べて記述する。comandはtargetを生成するコマンドで、次の行に記述します。複数のコマンドが必要なこともあり、その場合一行に一つずつコマンドを記述する。このとき注意するのはコマンドを記述する前にタブでインデントを行わなければならないことである。

コンパイルの流れ

getsについて

今回のプログラム、実はgetsが使われているためプログラム実行時にwarning: this program uses gets(), which is unsafe.というような警告文が表示される。これは長過ぎる行が入ったときバッファのオーバーを起こすことになる。

これを回避するためにはgetsをfgetsに置き換えれば良い。

asmプログラムの25行目を

```
while(fgets(num,sizeof(num),stdin) != NULL){  
書き換えれば警告文が出なくなる。
```

しかし、これではまた新たに問題が出る。それは出力結果で改行が一つ多くなることだ。これはfgetsが文字を打ったときの改行まで読み込むことにあるようだ。そこで26行目にnum[strlen(num)-1] = '\0';を追加すると問題点はなくなる。この意味は文字列終端文字一つ前の文字(改行)をNULLに置き換えるという意味だ。

[5] 感想

今回で前期の最後のプログラミングのレポートです。考えると長かったような短かったような感じですね。だけど結局、一つとして一日前に終わったレポートはなかったです、今回も今までと同様、いやそれ以上に切羽詰まりました。だって難しいんだもの、しかも手書きがメイン。きっと先生の授業でノートをとっていなければプログラム作成にはもっと時間がかかっていただろうとわかります。あ、それと今回の課題をやって気づいたんですが前回のレポートで自分、降順と昇順の意味を逆にとっていたようで……。とりあえず、今回はこれで終了。夏休みには復習をするようにがんばりたいところです。後期はJavaだとうかがっています、だからその予習もできるといいな……。

[6] 参考文献

C実践プログラミング	Steve Oualline 著
	望月 康司 監訳
	谷口 功 訳

初級C言語Q&A(9)	http://www.st.rim.or.jp/~phinloda/cqa/cqa9.html
-------------	---