

プログラミング I

REPORT#8

提出日	: 2012年 8月 2日 木曜日
所属	: 琉球大学工学部情報工学科
学籍番号	: 125722G
氏名	: 知念 辰明

1. 入力した正の整数を降順に並べ替えて出力するプログラムを作成せよ。
プログラムは個別にコンパイルし、make コマンドで実行すること。

a-0: Makefile の作成。

Makefile の中身▼

```
#
# RP8のためのメイクファイル的なものです。
#

cardinal.e:cardinal.o conv10.o select_sort.o print_num.o msg.o
    gcc -o cardinal.e cardinal.o conv10.o select_sort.o print_num.o msg.o

cardinal.o:cardinal.c
    gcc -c cardinal.c

conv10.o:conv10.c
    gcc -c conv10.c

select_sort.o:select_sort.c
    gcc -c select_sort.c

print_num.o:print_num.c
    gcc -c print_num.c

msg.o:msg.c
    gcc -c msg.c
```

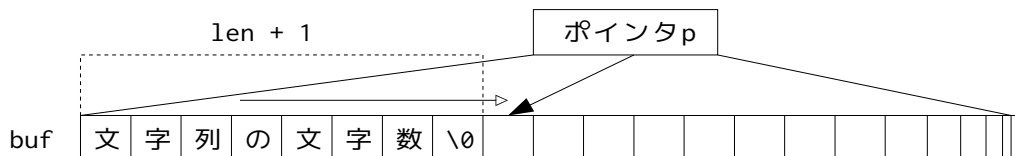
a-1: コードの穴を埋めつつ。

main 関数の含まれるファイルの高水準コードの全体▼

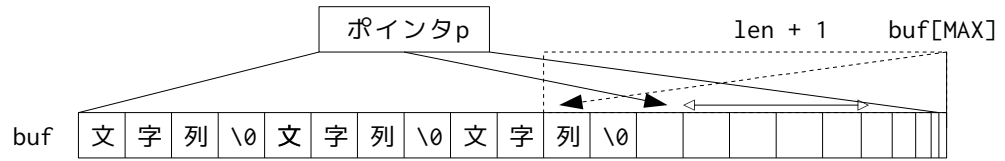
```
1  /*
2  program :cardinal.c
3  student-ID:125722G
4  author  :Tatsunori Chinen
5  data    :7/30/2012 (MON)
6  comments :Main program for List Struct
7  */
8  #include <stdio.h>                //ヘッダファイル読み込み
9  #include <string.h>
10 #define MAX 256                   //MAXという文字列を256に置換
11
12 void conv10(char **x, int *k, int n); //サブルーチンを宣言
13 void select_sort(int x[], int m[], int n);
14 void print_num(char *x[], int m[], int n);
15 void msg();
16
17 int main(){                        //メイン関数
18     char *dt[50], num[10], buf[MAX], *p=buf; //char型の配列やポインタを宣言
19     int n=0, len, i10[50], move[50]; //int型の変数や配列を宣言
20
21     puts("----- Input"); //文字を出力
22     while(gets(num) != NULL) { //配列numに文字列を取り込み, NULLが入力されるまでループ
23         len = strlen(num); //変数lenにnumに取り込まれた文字列の文字数を代入
24         if(p > buf+MAX-(len+1)) break; //pの値が"buf+MAX-(len+1)"より大きければbreak
25         strcpy(p, num); //num内の文字列をpが指しているアドレスを先頭に代入
26         dt[n] = p; //dt[n]にpが今指しているアドレスを代入
27         p+=len+1; //pを"len+1"(文字数+NULL分)だけ増加
28         n++; //nをインクリメント
29     }
30     puts("----- Result"); //文字を出力
31     conv10(dt, i10, n); //dt, i10, nを引数にサブルーチンconv10を呼び出し
32     select_sort(i10, move, n); //i10, move, nを引数にサブルーチンselect_sortを呼び出し
33     print_num(dt, move, n); //dt, move, nを引数にサブルーチンprint_numを呼び出し
34     msg(); //サブルーチンmsgを呼び出し
35
36     return(0);
37 }
```

考察▼

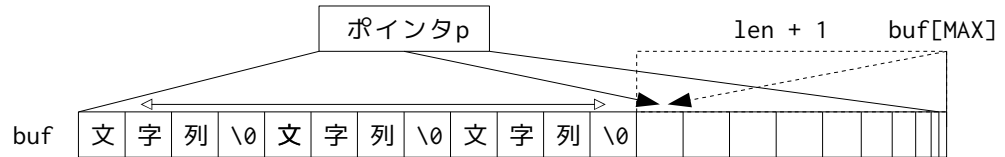
- 各ファイルの内、このファイル("cardinal.c")では各配列・ポインタ・変数の宣言、文字列の取り込み、各サブルーチンへの引数引渡し、呼び込みなどが行われている。
- 穴あき箇所のうち、同一ファイル内で完結している 24 行目, 27 行目について考察したい。
- 講義中では、char 型配列 buf[MAX]に格納されている文字列の文字数と、新しく格納される文字列の文字数によって文字列の取り込みを中止するために char 型ポインタ変数 *p を取り扱っていた。
- そこで 27 行目では、入力された文字列の文字数が格納されている変数 len に空白文字 "NULL" の分を含めた len+1 分ポインタが示すアドレスをずらしている。
- この動きを図示すると以下ようになる。



- pの値を踏まえて24行目を考えると、配列bufに"len+1"分の文字列が格納できる最低限のスペースが無いときにbreak、文字列の取り込みを中止する。
- このような動きになるには、pに格納されているアドレスが"buf+MAX-(len+1)"(配列bufの最後から数えて"文字数+1"分)だけ動かしただけのメモリのアドレスより大きいとき文字列の取り込みを中止すればよい。



"格納されようとしている文字列の文字数+1"分配列に残りが無ければ"真"



"格納されようとしている文字列の文字数+1"分配列に残りがあれば"偽"

サブルーチン conv10 の高水準コードの一部▼

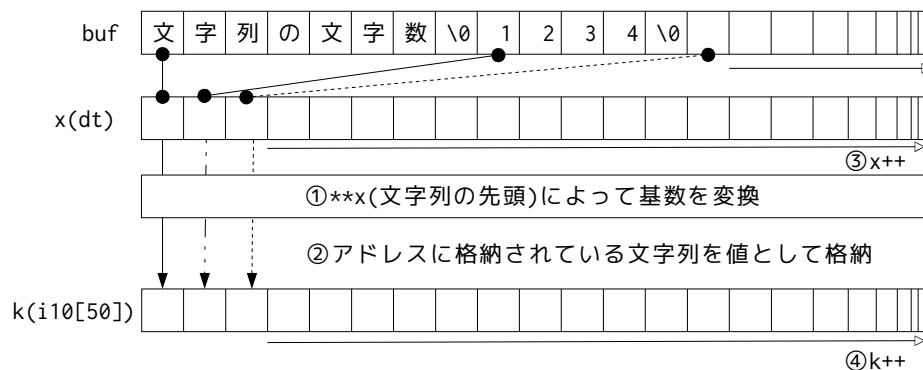
```

8 void conv10(char **x, int *k, int n){ //引数を"**, "*k", "n"で受け取り
9     while(n-- > 0){ //nをデクリメントした値が0以上であればループ
10        switch(**x){ //**x(*dt[50]が今指している文字)によって実行結果を変更
11            case '0' : //指している文字が"0"であった場合
12                sscanf(*x+1, "%o", k); //**xに1足したアドレスから続く文字列を8進数としてk(i10[50])に取り込む
13                break; //switch文を終了
14            case 'x' : //指している文字が"x"であった場合
15                sscanf(*x+1, "%x", k); //指している文字が"x"であった場合
16                break; //**xに1足したアドレスから続く文字列を16進数としてkに取り込む
17                //switch文を終了
18            default : //いずれにもあてはまらない場合
19                sscanf(*x, "%u", k); //**xから続く文字列を符号無し10進数としてkに取り込む
20                //switch文を終了
21        } //----switch文のブロック終了
22        x++; //**x(dt)のアドレスをインクリメント
23        k++; //kのアドレスをインクリメント
24    } //----while文のブロック終了
25 }

```

考察▼

- 各ファイルの内、このファイル("conv10.c")では格納されている文字列の先頭の文字によって読み込む際の基数を決定し読み込む処理を行う。
- ここで、サブルーチン"conv10.c"が引数として"cardinal.c"から受け取るのは、char型の2重ポインタ変数で表されるものと、int型ポインタ変数、及びint型変数である。
- "**x"はスイッチ文の条件として使われていることから既に何らかの値が入力されている。さらに、配列はポインタのように表記できることから引数として"char *dt[50]"を"**x"で受けている。
- "**dt[50]"には"cardinal.c"において読み込んだ文字列の先頭アドレスが入力されている。
- "**x"からスイッチ文を通して基数を判断したあと、文字列は"k"に格納されている。さらにkはint型ポインタ変数として表せるから、ここでは引数として"int i10[50]"を"*k"で受けている。
- 引数"int n"については、"cardinal.c"内の"n"、文字列の個数をそのまま"int n"で受けている。
- このルーチンの中での値の動きを図示すると以下のようになる。



サブルーチン select_sort の高水準コードの一部▼

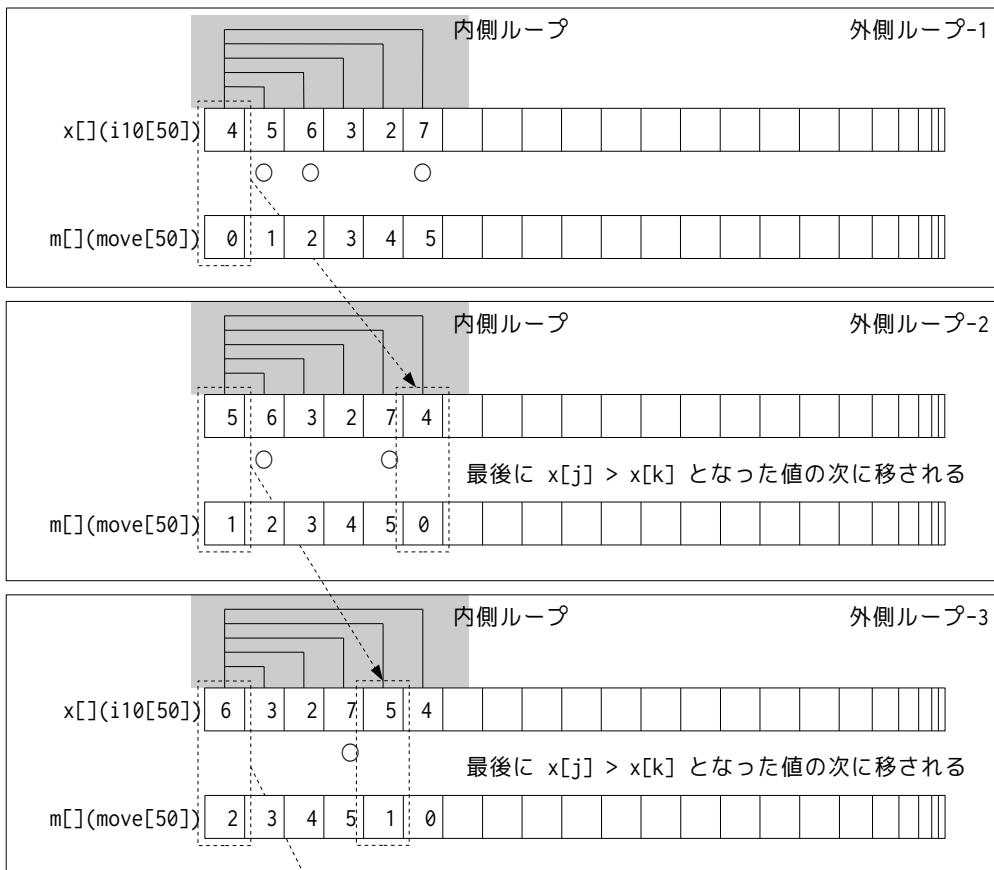
```

8 void select_sort(int x[], int m[], int n){
9     int i, j, k, w; //int型変数i,j,k,wを宣言
10    for(i=0; i<n; i++) m[i]=i; //”n”よりiが小さければ,m[i](move[50])の値にiの値を入力
11    //”n”(入力された値の数)の文だけループ
12    for(i=0; i<n-1; i++){ //”n-1”よりiが小さければループ
13        k=i; //kに今のiの値を入力
14        for(j=i+1; j<n; j++) //jに今のiの値+1を入れ,それがn以下なら繰り返しhをインクリメント
15            if( x[j] > x[k] ) k=j; //x[j](i10[j])の値がx[k](i10[k])の値より大きければ,kにjの値を代入
16        w = x[i]; //wにx[i](i10[i])の値を入力
17        x[i] = x[k]; //x[i](i10[i])にx[k]の値を入力
18        x[k] = w; //x[k](i10[k])にwの値を入力
19        w = m[i]; //wにm[i](move[i])の値を入力
20        m[i] = m[k]; //m[i](move[i])にm[k]の値を入力
21        m[k] = w; //m[k](move[k])にwの値を入力
22    }
23 }

```

考察▼

- 各ファイルの内,このファイル("select_sort.c")では,値に番号を割り振り,値の大小を判断し割り振った番号と共に入れ替えてソートを行う。
- ここで,サブルーチン"select_sort.c"が引数として"cardinal.c"から受け取るのは,int型の配列2つで表されるものとint型変数である。
- int型の配列2つの内,"int x[]"には最初から値が入っている為,"cardinal.c"内の"int i10[50]"があてはまる。
- 残された配列"int m[]"には,"cardinal.c"内の"int move[50]"があてはまる。
- "conv10.c"同様"int n"は"cardinal.c"の"int n",文字列の数を受けている。
- 14行目,15行目で値の大小を行うが,このときある値とその他の値をすべて比較する。
- 2重for文の中でループし,int整数j,kを用いて以下のような動きを値の数分繰り返す。
- 外側ループは"n-1",文字列の値より1つ少ない回数分ループする。



サブルーチン print_num の高水準コードの一部▼

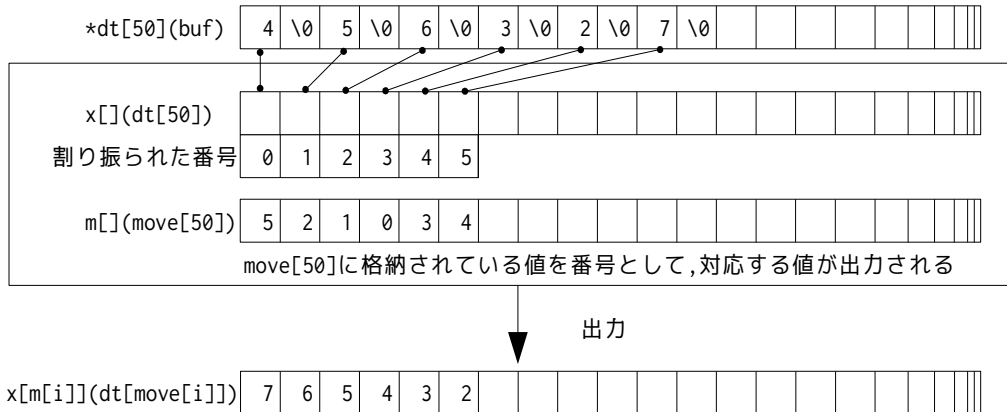
```

8 void print_num(char *x[], int m[], int n){ //引数を受け取る
9     int i; //変数iを宣言
10
11     for(i=0; i<n; i++){ //iを0で初期化,iがnより小さいときループ
12         puts(x[m[i]]); //x[m[i]](dt[move[i]])を出力
13     }
14 }

```

考察▼

- 各ファイルの内,このファイル("print_num.c")では,"select_sort.c"で割り振られた番号を用いてソートされた順通りに文字列を出力する。
- ここで,サブルーチン"print_num.c"が引数として"cardinal.c"から受け取るのは,char型のポインタ配列で表されるものと int 型の配列,int 型の整数である。
- puts()は文字列を出力する関数であるから,"*x[]"にあてはまるのは"cardinal.c"内の"*dt[50]"であり,添字として使われる int m[]には番号が格納された"move[50]"が,"int n"では文字列の数である"int n"を受ける。
- ここでの puts()関数の動きは以下のように図示される。



サブルーチン msg の高水準コードの一部▼

```

8 int msg(){
9     printf("#### Message from C #### By Chinen\n"); //メッセージを出力
10     return(0); //プログラムを終了
11 }

```

考察▼

- 各ファイルの内,このファイル("msg.c")では,C言語によるプログラムが終了したことを表すものである。

実行結果▼ ※入力の最後で"Control+D"を押すことで入力を終了している。

```

----- Input
warning: this program uses gets(), which is unsafe.
4
5
6
3
2
7
----- Result
7
6
5
4
3
2
#### Message from C #### By Chinen

```

実行結果(基数判断の動作確認)▼ ※入力の最後で"Control+D"を押すことで入力を終了している。

```

----- Input
warning: this program uses gets(), which is unsafe.
032
x32
32
12
012
X12
----- Result
x32
32
032
X12
12
012
#### Message from C #### By Chinen

```

XXX.あとがき。(反省・感想・参考)

a-1.参考サイト・文献。

- ・『C 実践プログラミング 第三版』(オーム社)
- ・『Open Lecture』
<http://www.osn.u-ryukyu.ac.jp/lecture/wiki/index.php?Open%20Lecture>

a-2.反省・感想。

ポインタ・配列やサブルーチン・for文などが入り交じって、今までのレポート・講義の復習にもなるレポートでした。言葉では説明がしづらいことも、図示することで分かりやすく説明でき、また同時により理解が深まりました。初めて習ったプログラミング言語として忘れないようにしたいです。

最後までご覧頂きありがとうございました！