プログラミング 1
Report#08

提出日：2009/07/23(木)
所属：工学部情報工学科
学籍番号：095736E
氏名：玉城　翔

課題１.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。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。

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

ソースプログラム：[sort1.c]

```
 1  /*
 2    Program    : sort1.c
 3    Comment    : 基数変換と整列処理
 4  */
 5
 6  #include <stdio.h>
 7  #include <string.h>
 8  #define  MAX 256
 9
10  void conv10(char **x, int *k, int n);
11  void select_sort(int x[], int m[], int n);
12  void print_num(char *x[], int m[], int n);
13  void msg();
14
15  int main(){
16    char *dt[50], num[64], buf[MAX], *p=buf;
17    int  n=0, len, i10[50], move[50];
18
19    puts("-------- Input");
20
21     while(gets(num) != NULL) {
22       len = strlen(num);
23
24       if(p > buf+MAX-(len+1) )  break;
25       strcpy(p, num);
26       dt[n] = p;
27       p += len+1;
28       n++;
29    }
30
31   puts("-------- Result");
32
33   conv10(dt, i10, n);
34
35   select_sort(i10, move, n);
36
37   print_num(dt, move, n);
38
39   msg();
40
41   return(0);
42
43  }
```

考察

main 関数での動作は読み込んだデータを、dt[]と buf[]に格納する動作が主になっている。
(２１行目)L21:gets 関数で num[]にデータを入力する。
L22:num[]に格納されているデータの長さの値を len に入れる。

L24:エラーを起こさない為の措置で、buf[]の容量を p が超えないような動作をしている。
L25:num[]のデータを p にコピーします。
L27:len+1 分の長さ分進み、それに+1 進みます。こうすることにより、次の入力に備えます。

num

X	4	2	1	\0	・・・・

↑ ↑

len の値(ここでは 4 となる)

buf

3	7	\0	X	4	2	1	\0	0	5	7	\0

ソースプログラム：[sort2.c]

```
 1  /*
 2    Program    : sort2.c
 3  */
 4  #include<stdio.h>
 5
 6  void conv10(char **x, int *k, int n){
 7
 8    while(n-- > 0){
 9
10      switch(**x){
11
12      case '0' :
13        sscanf(*x+1, "%o", k);
14        break;
15
16      case 'x' :
17      case 'X' :
18        sscanf(*x+1, "%x", k);
19        break;
20
21      default :
22        sscanf(*x, "%u", k);
23        break;
24
25      }
26
27      x++;
28      k++;
29
30    }
31
32  }
33
```

考察

conv10 関数は、dt[]に格納されているデータを１０進数・８進数・１６進数のどれかを判別し、それを値として k に格納します。
L8:while 文で n 回動作を繰り返します。
L10~25:switch 文で x(dt[])のデータを、１０進数・８進数・１６進数に判別します。
L12.16.17:８進数・１６進数は先頭がその進数を表す文字である為、その次のデータから sscanf で数値に変換し、k に格納します。

buf

3	7	\0	X	4	2	1	\0	0	5	7	\0

k(i10)

37	1057	47	・・・・

ソースプログラム：[sort3.c]

```
 1  /*
 2    Program    : sort3.c
 3  */
 4
 5  void select_sort(int x[], int m[], int n){
 6    int  i, j, k, w;
 7
 8    for(i=0; i<n; i++)  m[i]=i;
 9
10    for(i=0; i<n-1; i++){
11      k=i;
12
13      for(j=i+1; j<n; j++)
14
15        if( x[k] < x[j] )  k=j;
16
17      w    = x[i];
18      x[i] = x[k];
19      x[k] = w;
20
21      w    = m[i];
22      m[i] = m[k];
23      m[k] = w;
24
25    }
26
27  }
```

考察

select_sort 関数は、x[](i10)に格納されているデータをそれぞれ比べ、降順に並べ替えて、m[]に並べ替え方を記憶します。
L8:m[]に並べ替えの値を格納します。
L10~15:x[]内のデータの大小を比べていきます。
L17~23:比べたデータを並べ替えていきます。

ソースプログラム：[sort4.c]

```
 1  /*
 2    Program    : sort4.c
 3  */
 4
 5  void print_num(char *x[], int m[], int n){
 6    int  i;
 7
 8    for(i=0; i<n; i++){
 9      puts( x[m[i]] );
10
11    }
12
13  }
14
```

考察

print_num 関数は、dt[]のデータを並べ替えで行った時の並べ替えの順番 m[](move)にそって出力していく関数です。

ソースプログラム：[sort5.c]

```
/*
  Program    : sort5.c
*/
#include<stdio.h>

int msg(){
  printf("#### Message from C #### By Tamashiro \n");
  return(0);
}
```

メイクファイル：[makefile]

```
sort: sort1.o sort2.o sort3.o sort4.o sort5.o
	gcc -o  sort sort1.o sort2.o sort3.o sort4.o sort5.o

sort1.o: sort1.c
	gcc -c sort1.c

sort2.o: sort2.c
	gcc -c sort2.c

sort3.o: sort3.c
	gcc -c sort3.c

sort4.o: sort4.c
	gcc -c sort4.c

sort5.o: sort5.c
	gcc -c sort5.c

```

出力結果

```
-------- Input
37
x421
057
42
-------- Result
x421
057
42
37
#### Message from C #### By Tamashiro
```

考察

入力したデータが、きちんとソートされて降順の順に並び替えられて出力されていることが観て分かる。
gets 関数を使用している為か、しばしば警告の様なものが表示されていた。他にいい関数がないか調べておこうと思う。

課題２．リスト構造プログラムの動作を解析しなさい。

ソースプログラム：[list1.c]

```
/*
  Program    : list1.c
  Comment    : リスト構造
*/

#include <stdio.h>
```

```
 6  #include <stdlib.h>
 7
 8  #define FALSE 0
 9  #define TRUE  !FALSE
10
11  typedef struct Node{
12    int num;
13    struct Node *next_ptr;
14  }node;
15
16  node *start_ptr = NULL;
17
18  void ins(int idata){
19    node *p = start_ptr;
20
21    start_ptr = (node *)malloc(sizeof(node));
22    if (start_ptr == NULL) puts("Not enough memory!"), exit(0);
23
24    start_ptr->num = idata;
25    start_ptr->next_ptr = p;
26  }
27
28  int main(){
29    int idata;
30    node *p;
31
32    puts("Enter a sequence of integers:");
33    while(scanf("%d", &idata) == TRUE) ins(idata);
34
35    puts("In reverse order:");
36    for(p = start_ptr; p != NULL; p = p->next_ptr){
37      printf("%5d-", p->num);
38    }
39    puts("/end/");
40
41    return(0);
42  }
43
```

出力結果

```
1  Enter a sequence of integers:
2  123
3  58
4  190
5  34
6  In reverse order:
7     34-  190-   58-  123-/end/
```

考察

L10~14:struct node 型の構造体を作っている。Typedef により struct node 型に node という名前をつけている。

L15:node 型(struct 型)のポインタ start_ptr に NULL を入れる。

L26:main 関数から動作は始まる。

L30~31:scanf で idata に整数値を入力し、それを引数にして ins 関数にとばします。

L18:node 型 p に start_ptr を入れる。start_par には NULL が入っているので、p には NULL が入ることになる。

L19~21:malloc が node 型の大きさ分のメモリを確保し、start_ptr はそこの先頭アドレスを指す。malloc がメモリを確保出来なかった時、エラーメッセージを出す。

L22~23:start_ptr が指しているアドレスの num に idate の値を代入。next_ptr には p のアドレスを代入する。p は NULL を指しているため、つまり next_ptr は NULL を指す。

参考サイト

初心者のためのポイント学習C言語
http://www9.plala.or.jp/sgwr-t/index.html

e0857ポチギ13
http://www.ie.u-ryukyu.ac.jp/~e085713/home/report/syorui/c_rep6.pdf