

## [1] 課題内容

$nCr$ を再帰を用いて求め、Pascalの三角形を表示しプログラムの文法を箇条書きで解説し、プログラムの内容、アルゴリズムについて具体的に考察せよ。

## [2] プログラム

```
*****program1*****
/*
Program   : Pascal.c
Student-ID : 045713C
Author    : OSHIRO,Kazuya
Date      : 04/05/30
Comment   : 課題、Pascalの三角形プログラム
*/

#include <stdio.h>

#define N 12          /* Nという文字に12を定義する 繰り返しの数 */
long combi(int, int); /* combiをいう関数を定義するintという整数型パラメータを持つ */
/* int(4B),long(4B),で8バイトのメモリーがとられる */

int main()           /* main関数を呼び出す */
{
    int n, r, t;      /* 整数型のローカル変数n,r,tを宣言 */

    for (n=0; n<=N; n++){ /* 繰り返す回数がかわってくる0~12までの13回行う */
        for (t=0; t<(N-n)*3; t++) printf(" "); /* スペース */
        for (r=0; r<=n; r++) printf("%3ld ",combi(n,r)); /* 3桁の関数の値を出す */
        printf("¥n"); /* 改行を行う */
    }
    return(0);        /* main関数に0の値を返す */
}

long combi(int n, int r) /* combi関数の定義 */
{
    int i;              /* 整数型の変数iを宣言 */
    long p=1;           /* pを1に初期化する */
    for (i=1; i<=r; i++) p=p*(n-i+1)/i;
    return(p);          /* 計算結果pを呼び出し元へ返す */
}

*****
```

### [3] 実行結果

```
-----result1-----
[Kazuya-OSHIRO:~/prog1/report] j04013% ./Pascal
          1
         1 1
        1 2 1
       1 3 3 1
      1 4 6 4 1
     1 5 10 10 5 1
    1 6 15 20 15 6 1
   1 7 21 35 35 21 7 1
  1 8 28 56 70 56 28 8 1
 1 9 36 84 126 126 84 36 9 1
1 10 45 120 210 252 210 120 45 10 1
1 11 55 165 330 462 462 330 165 55 11 1
1 12 66 220 495 792 924 792 495 220 66 12 1

-rwxr-xr-x  1 j04013  j04013      11552 May 30 18:44 Pascal*
-rw-r--r--  1 j04013  j04013      1065 May 30 18:44 Pascal.c
-----
```

### [4] 考察

#### (1) プログラム文法

- ① `#include <stdin.h>` プリプロセッサのコマンドでファイル `stdio.h` (標準I/O) の内容をプログラム中に挿入する。
- ② `#define` ----- プリプロセッサのコマンドで値を定義するのに用いる。今回の場合記号定数 `N` を数値定数 `12` に置き換えている。
- ③ `combi(int, int)` ----- `combi` という関数を宣言する、`int` (整数型) のパラメータを持つ。
- ④ `main ()` ----- プログラム実行部 (本体) であり、この関数から他の関数を直接的、あるいは間接的に呼び出す。
- ⑤ `for` ----- `for` (初期化; 条件; 反復文) その後に本文を書く。条件に真である限り繰り返しプログラムを実行する。
- ⑥ `printf` ----- 変数指定によって記述された規則により、引数の数値や文字列を編集して、標準出力に出力する。
- ⑦ `return` ----- 関数に値を返す。

#### (2) プログラム内容

このプログラムは三角形の形を `for` 文と `printf` でスペースと改行を用いてつくり、その一つ一つに `combi` 関数の値 `p` を当てはめていくことにより、パスカルの三角形を表示するというプログラムである。

## ①各for文について

```
*****
for (n=0; n<=N; n++)
{
    for (t=0; t<(N-n)*3; t++) printf(" ");
    for (r=0; r<=n; r++) printf("%3ld ",combi(n,r));
    printf("¥n");
}
return(0);
*****
```

0	0C0
1	1C0 1C1
2	2C0 2C1 2C2
3	3C0 3C1 3C2 3C3

アのfor文はnを0に初期化、条件に合っているとき(真)下のfor文の処理をした後1を足す。これをnがN (つまりは12)になるまで繰り返す。このときnは上の図のように0~12の13個の値をとりそれは三角形の段(コンビネーションのn)を示している。条件に合わなくなったとき(偽)にはmain関数に値(0)を返しプログラムを終了する。

イのfor文はtを0に初期化、真ならばSPを出力し1を足す。nが0の時SPは36回出力され、n=1のとき33、n=2のとき30、n=3のとき27と続きこれにより始まりを階段のように表示できるようにになっている。ちなみにこのfor文をのぞくと結果が下のようになる。

```
-----result2-----
[Kazuya-OSHIRO:~/prog1/report] j04013% ./rep2
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
1 10 45 120 210 252 210 120 45 10 1
1 11 55 165 330 462 462 330 165 55 11 1
1 12 66 220 495 792 924 792 495 220 66 12 1

-rwxr-xr-x 1 j04013 j04013 11552 Jun 1 22:16 rep2*
-rw-r--r-- 1 j04013 j04013 849 Jun 1 22:16 rep2.c
-----
```

これより、直角三角形なることがわかった。

ウのfor文はtを初期化、条件はn以下、真ならば関数combi(int n, int r)から値を呼び出し出力するその後tに1を足す。イをウには条件にnが使われているためnによって繰り返す回数がかかる。条件に対し偽をしめた場合改行が行われるようになっている。

## ②combi関数の定義について

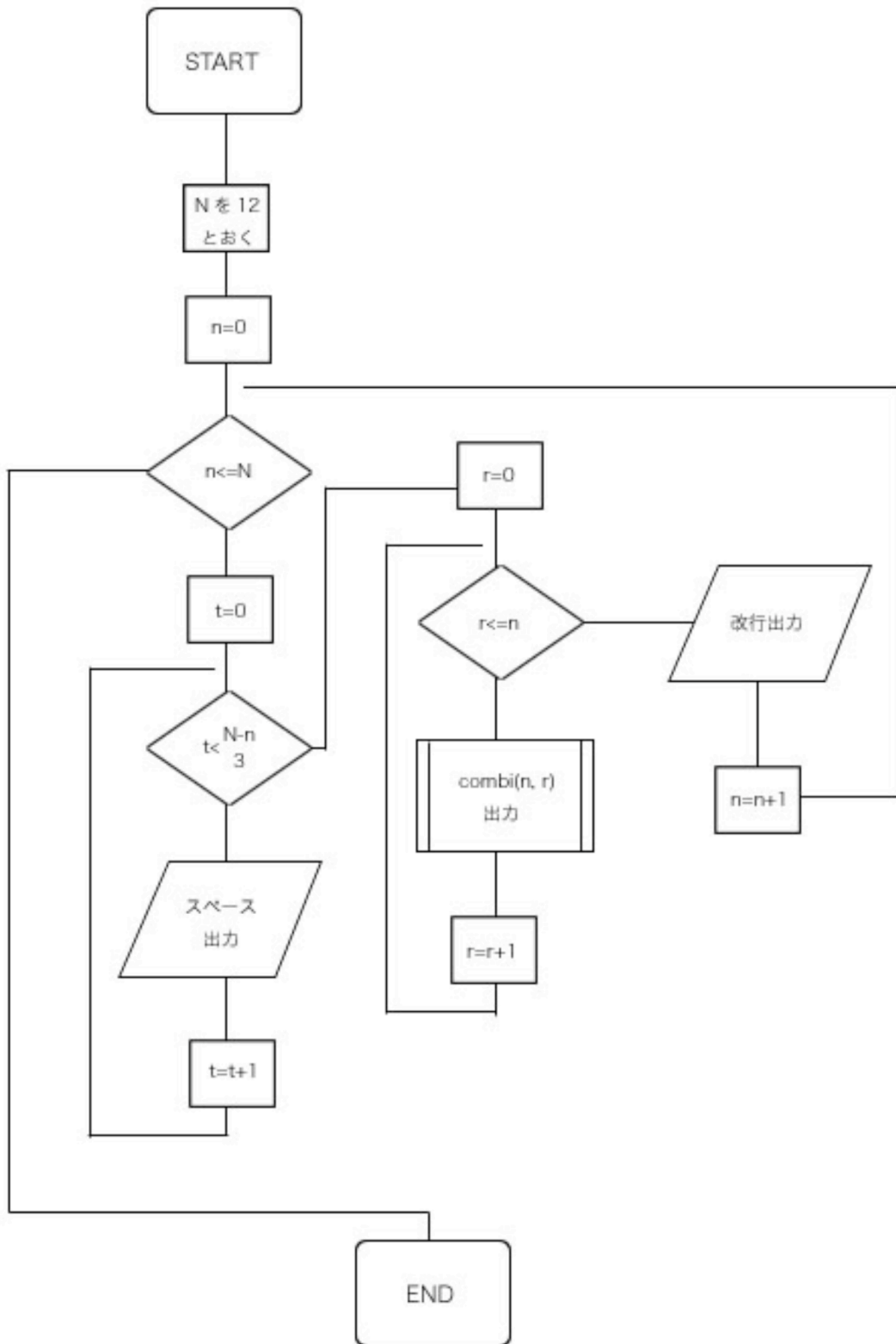
```
*****
long combi(int n, int r)
{
    int i;
    long p=1;
    for (i=1; i<=r; i++) p=p*(n-i+1)/i;
    return(p);
}
*****
```

$$\text{combi (int n, intr)} = nCr = \frac{nPr}{r!} = \frac{n(n-1)(n-2) \cdots (n-r+1)}{1*2*3* \cdots *r}$$

コンビネーションの定理より上の式のようになる $p=p*(n-i+1)/i$ はfor文が偽になるまでpを置き換えられてかけられるので結果、上の式と同等になる。

### (3)アルゴリズム

今回はフローチャートを用いた。



#### (4)プログラムの変更

##### ①逆三角形

今回のプログラムで、自分はパスカルの三角形を逆三角形にしようと試みた。まず、for文の考察で述べたように変数nは三角形の段を表しておりr、tにおいてもnの値によってかわるようなのでnをはじめにNと初期化し、一つずつ下げていく形にした。つまり(n=0; n<=N; n++)というところを(n=N; n<=0; n--)とだけ変更した。こうすることにより元の三角形では12段目であるところから処理を行うため逆三角形になる。この結果がこれである。

```
-----result3-----
[Kazuya-OSHIRO:~/prog1/report] j04013% ./Pascal-re
 1  12  66  220  495  792  924  792  495  220  66  12  1
  1  11  55  165  330  462  462  330  165  55  11  1
   1  10  45  120  210  252  210  120  45  10  1
    1  9  36  84  126  126  84  36  9  1
     1  8  28  56  70  56  28  8  1
      1  7  21  35  35  21  7  1
       1  6  15  20  15  6  1
        1  5  10  10  5  1
         1  4  6  4  1
          1  3  3  1
           1  2  1
            1  1
             1

-rwxr-xr-x  1 j04013  j04013      11552 Jun  2 18:19 Pascal-re*
-rw-r--r--  1 j04013  j04013      1150 Jun  2 18:18 Pascal-re.c
-----
```

##### ②,再帰を用いたプログラム

次に、再帰関数を用いてパスカルの三角形を表示することを試みました。

```
*****program2*****
/*
Program      : Pascal-return.c
Student-ID   : 045713C
Author       : OSHIRO,Kazuya
Date         : 04/06/02
Comment      : 課題、Pascalの三角形プログラム(再帰)
*/

#include <stdio.h>

#define N 12 /* Nという文字に12をわりあてる 繰り返しの数 */
long combi(int, int); /* combiをいう関数を定義するintという~を持つ(p65) */
/* int(4B),long(4B),で8バイトのメモリーがとられる */
int main() /* */
{
    int n, r, t; /* 整数型の変数n,r,tを宣言 */

    for (n=N; n>=0; n--){
        for (t=0; t<(N-n)*3; t++) printf(" "); /* スペース */
        for (r=0; r<=n; r++) printf("%3ld ",combi(n,r)); /* 3桁の関数の値を出す */
    }
}
```

```

    printf("%n");          /* 繰り返す回数が変わってくる0~12までの13回行う */
}
return(0);                /* mainに0の値を返す */
}

long combi(int n, int r) /* combi関数の定義 */
{
    int c, c1, c2;
    if(r==0||r==n)return(1); /* 条件が真なら値1を返す */
    /* else */
    c1=combi(n-1, r);        /* c1にn-1Crを代入 */
    c2=combi(n-1, r-1);     /* c2にn-1Cr-1を代入 */
    c=c1+c2;                /* コンビネーションの定義nCr=n-1Cr+n-1Cr-1 */
    return(c);              /* コンビネーションの値cを返す */
}

```

\*\*\*\*\*  
 再帰関数とは関数自体が関数を呼び出すものである。ここでのcおよびc1,c2はrが0もしくはnになるまで何度も呼び出され処理される。このプログラムの結果は下のようになる。出力される数値はかわらないため元の逆三角形と同じようになる。

```

-----
[Kazuya-OSHIRO:~/prog1/report] j04013% ./Pascal-return
 1  12  66  220  495  792  924  792  495  220  66  12  1
  1  11  55  165  330  462  462  330  165  55  11  1
   1  10  45  120  210  252  210  120  45  10  1
    1  9  36  84  126  126  84  36  9  1
     1  8  28  56  70  56  28  8  1
      1  7  21  35  35  21  7  1
       1  6  15  20  15  6  1
        1  5  10  10  5  1
         1  4  6  4  1
          1  3  3  1
           1  2  1
            1  1
             1

```

```

-rwxr-xr-x  1 j04013  j04013      11552 Jun  2 16:38 Pascal-return*
-rw-r--r--  1 j04013  j04013      1150  Jun  2 16:47 Pascal-return.c

```

## [5] 反省・感想

はじめこのプログラムを見たとき何を言っているのかさっぱりだったが、ちゃんとやれば少しずつだわわかるようになり、理解できるようになったことがうれしかったです。前回の反省を生かせず今回も始めるのが遅かったため終わらすのはギリギリでした。その点は本当に気をつけていきたいと思いました。今回は関数の使い方が少しわかったと思います。

## 今回使用したコンパイラ

gccコンパイラ

## 参考文献

C実践プログラミング 第3版 (Steve Oualline)

Cプログラミング (ウイネット)

アルゴリズム 入門編 (ウイネット)

IT用語辞典 e-Words (<http://e-words.jp/>)