

コンパイルの例

PASCAL ソースコード

```
program fibo(input, output);
var N: integer;

procedure nacci(fn1, fn2, n : integer);
var fn : integer;
begin
    fn := fn1 + fn2;
    writeln('F_', n, ' = ', fn);
    if n < N then
    begin
        nacci(fn, fn1, n+1)
    end
end;

begin
    N := 23;
    writeln('Fibonacci Numbers');
    writeln('F_1 = 1');
    writeln('F_2 = 1');
    nacci(1,1,3)
end.
```

対応する CASL のコード

```
1  CASL    START  _BEGIN
2  _OUTBUF DS    256
3  _BEGIN  LAD    GR6, 0
4          LAD    GR7, _OUTBUF
5          JUMP   MAIN
6  ::: nacci begin :::
7  _NACCI  PUSH   0, GR5
8          LD     GR5, GR8
9          PUSH   0
10 ::: begin assign :::
11         LD     GR1, 4, GR5
12         PUSH   0, GR1
13         LD     GR1, 3, GR5
14         LAD    GR2, 0, GR1
15         POP    GR1
16         ADDA   GR1, GR2
17         ST     GR1, -1, GR5
18 ::: end assign :::
19         LAD    GR2, ST0
20         LAD    GR1, 2
21         CALL   WRTSTR
22         LD     GR1, 2, GR5
23         LD     GR2, GR1
24         CALL   WRTINT
25         LAD    GR2, ST1
26         LAD    GR1, 3
27         CALL   WRTSTR
28         LD     GR1, -1, GR5
29         LD     GR2, GR1
30         CALL   WRTINT
31         CALL   WRTLN
32         LD     GR1, 2, GR5
33         PUSH   0, GR1
34         LD     GR1, _N
```

```

35      LAD      GR2, 0, GR1
36      POP      GR1
37      CPA      GR1, GR2
38      LAD      GR1, 1
39      JMI      L0
40      LAD      GR1, 0
41 L0    LD       GR1, GR1
42      JZE      L1
43      LD       GR1, -1, GR5
44      PUSH     0, GR1
45      LD       GR1, 4, GR5
46      PUSH     0, GR1
47      LD       GR1, 2, GR5
48      PUSH     0, GR1
49      LAD      GR1, 1
50      LAD      GR2, 0, GR1
51      POP      GR1
52      ADDA     GR1, GR2
53      PUSH     0, GR1
54      CALL     _NACCI
55      POP      GRO
56      POP      GRO
57      POP      GRO
58 L1    NOP
59      LAD      GR8, 1, GR5
60      LD       GR5, 0, GR5
61      RET
62      ;;; nacci end ;;;
63 ERR   LD       GR2, GR1
64      CALL     WRTINT
65      CALL     WRTLN
66      LAD      GR2, ERR_MSG
67      LD       GR1, ERR_LEN
68      CALL     WRTSTR
69      CALL     WRTLN
70      LAD      GR8, #FEFF

```

```

71          RET
72 MAIN    NOP
73          LD      GR5, GR8
74  ::: begin assign :::
75          LAD     GR1, 23
76          ST      GR1, _N
77  ::: end assign :::
78          LAD     GR2, ST2
79          LAD     GR1, 17
80          CALL    WRTSTR
81          CALL    WRTLN
82          LAD     GR2, ST3
83          LAD     GR1, 7
84          CALL    WRTSTR
85          CALL    WRTLN
86          LAD     GR2, ST4
87          LAD     GR1, 7
88          CALL    WRTSTR
89          CALL    WRTLN
90          LAD     GR1, 1
91          PUSH    0, GR1
92          LAD     GR1, 1
93          PUSH    0, GR1
94          LAD     GR1, 3
95          PUSH    0, GR1
96          CALL    _NACCI
97          POP     GRO
98          POP     GRO
99          POP     GRO
100 END     NOP
101          RET
102  ::: GLOBAL :::
103  _N      DS      1
104  ::: CONST :::
105
106  STO     DC      'F_'

```

```

107 ST1    DC    ' = '
108 ST2    DC    ' Fibonacci Numbers'
109 ST3    DC    ' F_1 = 1'
110 ST4    DC    ' F_2 = 1'
111 ERR_MSG DC    'error: array index is out of bounds'
112 ERR_LEN DC    35
113        END
114 ;=====
115 ; MULT: 掛け算を行うサブルーチン
116 ; GR1 * GR2 -> GR2
117 MULT    START
118        PUSH    0, GR1 ; GR1 の内容をスタックに退避
119        PUSH    0, GR3 ; GR3 の内容をスタックに退避
120        PUSH    0, GR4 ; GR4 の内容をスタックに退避
121        LAD     GR3, 0 ; GR3 を初期化
122        LD      GR4, GR2
123        JPL     LOOP
124        XOR     GR4,=#FFFF
125        ADDA    GR4,=1
126 LOOP    SRL     GR4, 1
127        JOV     ONE
128        JUMP    ZERO
129 ONE     ADDL   GR3, GR1
130 ZERO    SLL     GR1, 1
131        AND     GR4, GR4
132        JNZ     LOOP
133        CPA     GR2,=0
134        JPL     END
135        XOR     GR3,=#FFFF
136        ADDA    GR3,=1
137 END     LD      GR2, GR3
138        POP     GR4
139        POP     GR3
140        POP     GR1
141        RET
142        END

```

```

143
144 ;=====
145 ; DIV 割り算を行うサブルーチン
146 ; GR1 / GR2 -> 商は GR2, 余りは GR1
147 DIV    START
148        PUSH    0, GR3 ; GR3 の内容をスタックに退避
149        PUSH    0, GR4 ; GR4 の内容をスタックに退避
150        PUSH    0, GR1 ; GR1 (被除数)をスタックに退避
151        PUSH    0, GR2 ; GR2 (除数)をスタックに退避
152        CPA     GR1, =0
153        JPL     ZYOSU1
154        XOR     GR1, =#FFFF
155        ADDA    GR1, =1
156 ZYOSU1 CPA     GR2, =0
157        JPL     KEISAN
158        XOR     GR2, =#FFFF
159        ADDA    GR2, =1
160 KEISAN LAD     GR3, 1
161        SUBL    GR4, GR4
162 PRE    CPL     GR1, GR2
163        JZE     LOOP
164        JMI     LOOP
165        SLL     GR3, 1
166        SLL     GR2, 1
167        JPL     PRE
168 LOOP  CPL     GR1, GR2
169        JMI     CHECK
170        OR      GR4, GR3
171        SUBL    GR1, GR2
172        JZE     POST
173 CHECK SRL     GR2, 1
174        SRL     GR3, 1
175        JNZ     LOOP
176 POST  LD      GR2, GR4
177        POP     GR3 ; 被除数が 0 ならば GR2 を -1 倍
178        CPA     GR3, =0

```

```

179          JPL      ZYOSU2
180          XOR      GR2,=#FFFF
181          ADDA     GR2,=1
182 ZYOSU2 POP      GR3 ; 除数が 0 ならば GR2 を -1 倍
183          CPA      GR3,=0
184          JPL      END
185          XOR      GR2,=#FFFF
186          ADDA     GR2,=1
187 END      POP      GR4
188          POP      GR3
189          RET
190          END
191
192 ;=====
193 ; 入力装置から数値データを読み込み,
194 ; その内容を GR2 が指すアドレスに格納するサブルーチン
195 RDINT   START
196          PUSH     0,GR1 ; GR1 の内容をスタックに退避
197          PUSH     0,GR3 ; GR3 の内容をスタックに退避
198          PUSH     0,GR4 ; GR4 の内容をスタックに退避
199          PUSH     0,GR5 ; GR5 の内容をスタックに退避
200          PUSH     0,GR6 ; GR6 の内容をスタックに退避
201          LD       GR5,GR2 ; GR2 が指す番地を GR5 にコピー
202          LD       GR2,=0 ; GR2 を初期化
203          LD       GR3,=0 ; GR3 を初期化
204          IN       INAREA,INLEN ; 入力を受け取る
205
206          ; 入力が null かどうかのチェック
207          CPA      GR3,INLEN
208          JZE      ERROR
209
210          ; 最初の文字が '-' かどうかのチェック
211          LD       GR4,INAREA,GR3
212          LAD      GR3,1,GR3
213          LD       GR6,GR4 ; GR6 に入力された先頭の文字を保存
214          CPL      GR4,=#002D ; '-' かどうか

```

```

215      JZE      LOOP
216      CPL      GR4,='0' ; 数値かどうかのチェック
217      JMI      ERROR
218      CPL      GR4,='9'
219      JPL      ERROR
220      XOR      GR4,#0030 ; 数値だったら変換
221      ADDA     GR2,GR4
222
223      ; 「すでに読み込んだ数値を 10 倍して,
224      ; 新しく読み込んだ数値と足す。」を繰り返す
225 LOOP  CPA     GR3,INLEN
226      JZE      CODE ; 入力された文字数と GR3 が同じであればループを抜ける
227      LD       GR1,=10
228      CALL     MULT ; GR2 の値を 10 倍する
229      LD       GR4,INAREA,GR3
230      CPL      GR4,='0' ; 数値かどうかのチェック
231      JMI      ERROR
232      CPL      GR4,='9'
233      JPL      ERROR
234      XOR      GR4,#0030 ; GR4 の内容を数値に変換
235      ADDA     GR2,GR4 ; GR2 に GR1 の内容を足す
236      LAD      GR3,1,GR3; GR3 (ポインタ)をインクリメント
237      JUMP     LOOP
238
239      ; 最初の文字が '-' であった場合は, -1 倍する
240 CODE  CPL      GR6,#002D
241      JNZ      END
242      XOR      GR2,#FFFF
243      LAD      GR2,1,GR2
244      JUMP     END
245
246      ; エラーを出力する
247 ERROR OUT      ERRSTR,ERRLEN
248
249 END   ST       GR2,0,GR5 ; GR2 の内容を GR5 が指す番地に格納する
250      LD       GR2,GR5 ; GR5 が指す番地を GR2 に戻す

```



```

251      POP      GR6
252      POP      GR5
253      POP      GR4
254      POP      GR3
255      POP      GR1
256      RET
257
258  ERRSTR  DC      'illegal input'
259  ERRLEN  DC      13
260  INAREA  DS      6
261  INLEN   DS      1
262      END
263
264  ;=====
265  ; 入力装置から文字を読み込み,
266  ; その内容を GR2 が指すアドレスに格納するサブルーチン
267  RDCH    START
268      IN      INCHAR, INLEN
269      LD      GR1, INCHAR
270      ST      GR1, 0, GR2
271      RET
272  INCHAR  DS      1
273  INLEN   DS      1
274      END
275
276  ;=====
277  ; 入力装置から, GR1 の文字数を読み込む.
278  ; 読み込んだ文字列は, GR2 が指すアドレスから順に格納される
279  RDSTR   START
280      PUSH   0, GR3 ; GR3 の内容をスタックに退避
281      PUSH   0, GR4 ; GR4 の内容をスタックに退避
282      PUSH   0, GR5 ; GR5 の内容をスタックに退避
283      LAD    GR4, 0 ; GR4 を初期化
284      IN      INSTR, INLEN
285  LOOP    CPA    GR4, GR1
286      JZE    END ; GR1 で指定された文字数を超えたら終わり

```

```

287         CPA      GR4, INLEN
288         JZE      END ; 入力された文字数を超えたら終わり
289         LD       GR5, GR2
290         ADDA     GR5, GR4 ; 文字の格納先番地を計算
291         LD       GR3, INSTR, GR4
292         ST       GR3, 0, GR5
293         LAD      GR4, 1, GR4
294         JUMP     LOOP
295 END      POP      GR5
296         POP      GR4
297         POP      GR3
298         RET
299 INSTR   DS      256
300 INLEN   DS      1
301         END
302
303 ;=====
304 ; 入力装置からの文字列を改行まで読み飛ばすサブルーチン
305 RDLN    START
306         IN       INAREA, INLEN
307         RET
308 INAREA  DS      256
309 INLEN   DS      1
310         END
311
312 ;=====
313 ; GR2 の内容 (数値データ) を出力装置に書き出すサブルーチン
314 ; このサブルーチンが呼ばれたとき,
315 ; GR7 には, 出力用番地の先頭アドレスが,
316 ; GR6 には, 現在出力用番地に入っている文字数が,
317 ; それぞれ格納されている.
318 WRTINT  START
319         PUSH     0, GR1 ; GR1 の内容をスタックに退避
320         PUSH     0, GR2 ; GR2 の内容をスタックに退避
321         PUSH     0, GR3 ; GR3 の内容をスタックに退避
322         PUSH     0, GR2 ; 数値データをもう一度スタックに退避

```

```

323      LD      GR3,=0 ; GR3 はインデックスとして用いる
324
325      ; 数値データが負数である場合は, 正の数に変換
326      CPA      GR2,=0
327      JPL      LOOP1
328      XOR      GR2,=#FFFF
329      ADDA     GR2,=1
330
331      ; 数値データを変換しながら, バッファに格納
332 LOOP1  LD      GR1, GR2
333      LD      GR2,=10
334      CALL     DIV
335      XOR      GR1,=#0030
336      ST      GR1, BUFFER, GR3
337      LAD     GR3, 1, GR3
338      CPA      GR2,=0
339      JNZ     LOOP1
340
341      ; 数値データが負数であれば, '-' を追加
342      POP     GR2
343      CPA      GR2,=0
344      JZE     LOOP2
345      JPL     LOOP2
346      LD      GR1,='- '
347      ST      GR1, BUFFER, GR3
348      LAD     GR3, 1, GR3
349
350      ; BUFFER を逆順にたどりながら, 出力用バッファに格納
351 LOOP2  LAD     GR3, -1, GR3
352      LD      GR1, BUFFER, GR3
353      LD      GR2, GR7
354      ADDA     GR2, GR6
355      ST      GR1, 0, GR2
356      LAD     GR6, 1, GR6
357      CPA      GR3,=0
358      JNZ     LOOP2

```

```

359
360 END      POP      GR3
361          POP      GR2
362          POP      GR1
363          RET
364 BUFFER DS      6
365          END
366
367 ;=====
368 ; GR2 の内容（文字）を出力装置に書き出すサブルーチン
369 ; このサブルーチンが呼ばれたとき,
370 ; GR7 には, 出力用番地の先頭アドレスが,
371 ; GR6 には, 現在出力用番地に入っている文字数が,
372 ; それぞれ格納されている.
373 WRTCH  START
374          PUSH     0, GR1 ; GR1 の内容をスタックに退避
375          LD       GR1, GR7
376          ADDA    GR1, GR6 ; GR1 に次の文字を格納する番地を代入
377          ST      GR2, 0, GR1
378          LAD     GR6, 1, GR6
379          POP     GR1
380          RET
381          END
382
383 ;=====
384 ; GR2 の指すメモリ番地から, 長さ GR1 の文字列を出力装置に書き出すサブルー
チン
385 ; このサブルーチンが呼ばれたとき,
386 ; GR7 には, 出力用番地の先頭アドレスが,
387 ; GR6 には, 現在出力用番地に入っている文字数が,
388 ; それぞれ格納されている.
389 WRTSTR START
390          PUSH     0, GR3 ; GR3 の内容をスタックに退避
391          PUSH     0, GR4 ; GR4 の内容をスタックに退避
392          PUSH     0, GR5 ; GR5 の内容をスタックに退避
393          LAD     GR3, 0 ; GR3 は制御変数として用いる

```

```

394 LOOP   CPA    GR3, GR1
395       JZE    END
396       LD     GR4, GR2
397       ADDA   GR4, GR3 ; 出力する文字の格納番地を計算
398       LD     GR5, 0, GR4 ; 出力する文字をレジスタにコピー
399       LD     GR4, GR7
400       ADDA   GR4, GR6 ; 出力先の番地を計算
401       ST     GR5, 0, GR4 ; 出力装置に書き出し
402       LAD    GR3, 1, GR3
403       LAD    GR6, 1, GR6
404       JUMP   LOOP
405 END    POP    GR5
406       POP    GR4
407       POP    GR3
408       RET
409       END
410
411 ;=====
412 ; 改行を出力装置に書き出すサブルーチン
413 ; 実質的には, GR7 で始まるアドレス番地から長さ GR6 の文字列を出力する
414 WRTLN  START
415       PUSH   0, GR1
416       PUSH   0, GR2
417       PUSH   0, GR3
418       ST     GR6, OUTLEN
419       LAD    GR1, 0
420
421 LOOP   CPA    GR1, OUTLEN
422       JZE    END
423
424       LD     GR2, GR7
425       ADDA   GR2, GR1
426       LD     GR3, 0, GR2
427       ST     GR3, OUTSTR, GR1
428       LAD    GR1, 1, GR1
429       JUMP   LOOP

```

```
430
431 END      OUT      OUTSTR, OUTLEN
432         LAD      GR6, 0 ; 文字列を出力して, GR6 を初期化
433         POP      GR3
434         POP      GR2
435         POP      GR1
436         RET
437 OUTSTR DS      256
438 OUTLEN DS       1
439         END
440
```