

Risa/Asir ドリル 8 章 “関数”, 12 章 “再帰呼び出し”
を C 言語で.

Risa/Asir ドリル 2022, p.249, 24 章 “Risa/Asir プ
ログラマのための C 言語入門” も参照.

コード 1: 和, C 言語版 nsum.c

```
1 #include <stdio.h>
2 int nsum(int N) {
3     int S;
4     int I;
5     S=0;
6     for (I=1; I<=N; I++) {
7         S = S+I;
8     }
9     printf("S_in_nsum=%d\n",S);
10    return S;
11 }
12 int main() {
13     int S,N;
14     S=3;
15     N=100;
16     printf("S=%d\n",S);
17     printf("nsum(%d)=%d\n",N,nsum(N));
18     printf("S=%d\n",S);
19 }
```

コード 2: 和, asir 版 nsum.rr

```
1 def nsum(N) {
2     S=0;
3     for (I=1; I<=N; I++) {
4         S = S+I;
5     }
6     printf("S_in_nsum=%a\n",S);
7     return S;
8 }
9 S=3$
10 N=100$
11 printf("S=%a\n",S)$
12 printf("nsum(%a)=%a\n",N,nsum(N))$
13 printf("S=%a\n",S)$
14 end$
```

コード 3: ベクトルの和, C 言語版 vecadd.c

```
1 #include <stdio.h>
2 #define N 3
3 void vecadd(int A[N],int B[N],int C[N]) {
4     int I;
5     for (I=0; I<N; I++) {
6         C[I]=A[I]+B[I];
7     }
8 }
9 int main() {
10     int A[N] = {1,2,3};
11     int B[N] = {3,2,1};
12     int C[N];
13     int i;
14     vecadd(A,B,C);
15     printf("C=");
16     for (i=0; i<N; i++) printf("%d_",C[i]);
17     printf("\n");
18 }
```

コード 4: ベクトルの和 asir 版 vecadd.rr

```
1 def vecadd(A,B,C) {
2     N=length(A);
3     for (I=0; I<N; I++) {
4         C[I]=A[I]+B[I];
5     }
6 }
7
8 A=newvect(3,[1,2,3]);
9 B=newvect(3,[3,2,1]);
10 C=newvect(3);
11 vecadd(A,B,C);
12 printf("C=%a\n",C)$
13
14 def vecadd2(A,B) {
15     N=length(A);
16     C=newvect(N);
17     for (I=0; I<N; I++) {
18         C[I]=A[I]+B[I];
19     }
20     return C;
21 }
22 end$
```

コード 5: 再帰による gcd, C 言語版 gcd-r.c

```
1 #include <stdio.h>
2 int gcd_r(int A,int B) {
3     int T;
4     int R;
5     if (B>A) {
6         T = A; A = B; B=T;
7     }
8     R=A%B;
9     if (R==0) return B;
10    else return gcd_r(B,R);
11 }
12
13 /* Ax+By=D, A >= B の整数解の一つを見つける */
14 /* を言語で書くにはメモリ管理を理解していないと
15    z_syzC1 じゃないので初級としてはやらない
16 */
17
18 int main() {
19     printf("%d\n",gcd_r(128,24));
20 }
```

コード 6: 再帰による gcd, asir 版 gcd-r.rr

```
1 def gcd_r(A,B) {
2     if (B>A) {
3         T = A; A = B; B=T;
4     }
5     R=A%B;
6     if (R==0) return B;
7     else return gcd_r(B,R);
8 }
9
10 /* Ax+By=D, A >= B の整数解の一つを見つける */
11 def z_syz(A,B,D) {
12     if (B==0) {
13         if (D % A != 0) return 0; // no
14         solution
15         else return [D/A,0];
16     }
17     Q=idiv(A,B);
18     R=A%B;
19     S=z_syz(B,R,D);
20     if (S == 0) return 0;
21     return [S[1],S[0]-Q*S[1]];
22 }
```

```

23 gcd_r(128,24);
24 z_syz(128,24,16);
25 z_syz(128,24,7);

```

コード 7: C 曲線, C 言語版 ccurve-r.c

```

1 #include <stdio.h>
2 #include <math.h>
3 #include "glib5.h"
4 double CX;
5 double CY;
6 void lineto(double T,double R) {
7     extern double CX;
8     extern double CY;
9     double X;
10    double Y;
11    X=CX + R*cos(T);
12    Y=CY + R*sin(T);
13    glib_line(CX,CY,X,Y);
14    CX=X; CY=Y;
15 }
16 void moveto(double X,double Y) {
17     extern double CX;
18     extern double CY;
19     CX=X; CY=Y;
20 }
21
22 int ccurve2(double T,double R,int N) {
23     printf("T=%lf, R=%lf, N=%d\n",T,R,N);
24     if (N<2) {
25         lineto(T,R); return(0);
26     }
27     ccurve2(T+3.1415/4,R/sqrt(2),N-1);
28     ccurve2(T-3.1415/4,R/sqrt(2),N-1);
29     return 0;
30 }
31 int main() {
32     glib_open();
33     glib_loop();
34 }
35 void glib_draw() {
36     glib_window(-10,-10,15,15);
37     glib_clear();
38     glib_color(NULL,0xff0000);
39     moveto(0,0);
40     ccurve2(0,10,9);
41     glib_flush();
42 }

```

```

1 import("glib3.rr")$
2 extern CX$
3 extern CY$
4 def lineto(T,R) {
5     extern CX;
6     extern CY;
7     X=deval(CX + R*cos(T));
8     Y=deval(CY + R*sin(T));
9     glib_line(CX,CY,X,Y);
10    CX=X; CY=Y;
11 }
12 def moveto(X,Y) {
13     extern CX;
14     extern CY;
15     CX=X; CY=Y;
16 }
17 def ccurve2(T,R,N) {
18     print([T,R,N]);
19     if (N<2) {
20         lineto(T,R); return(0);
21     }
22     ccurve2(T+@pi/4,R/2^(1/2),N-1);
23     ccurve2(T-@pi/4,R/2^(1/2),N-1);
24 }
25 def main(N) {
26     glib_window(-10,-10,15,15);
27     glib_clear();
28     moveto(0,0);
29     ccurve2(0,10,N);
30     glib_flush();
31 }
32 end$

```

実行方法:

```
emacs ccurve-r.c &
```

でソースコードを入力して, shell で下記のようにコンパイル, 実行.

```
cc ccurve-r.c glib5.c -lX11 -lm
./a.out
```

glib5.c, glib5.h は講義のページからダウンロード. -l で C 言語のライブラリを指定. -lX11 は glib5.c ライブラリを用いて graphic を描く時に必要. -lm は sin, cos などを使うときに必要.

コード 8: C 曲線, asir 版 ccurve-r.rr
