

Risa/Asir ドリル 7 章 “ユークリッドの互除法と計算量” 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: 素因数分解, C 言語版

```
1 #include <stdio.h>
2 int prime_factorization(int N) {
3     int K;
4     K = 2;
5     printf("N=%d:", N);
6     while (N >= 2) {
7         if (N % K == 0) {
8             N = N/K;
9             printf("%d,", K);
10        }else{
11            K = K+1;
12        }
13    }
14    printf("\n");
15 }
16 int main() {
17     prime_factorization(60);
18     prime_factorization(34359738368);
19     prime_factorization(2147483650);
20 }
```

実行方法:

```
emacs prime.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 prime.c
./a.out
```

コード 2: 素因数分解 asir 版

```
1 def prime_factorization(N) {
2     K = 2;
3     printf("N=%a:", N);
4     while (N >= 2) {
5         if (N % K == 0) {
6             N = idiv(N, K);
7             print(K, 0); print(", ", 0);
8         }else{
9             K = K+1;
10        }
11    }
12    print("\n");
13 }
14
15 prime_factorization(60);
16 prime_factorization(34359738368);
17 prime_factorization(2147483650);
18 /* 34359738368=2^35 */
19 /* 2147483650=2^31+2 */
20 end$
```

コード 3: gcd 計算 C 言語版

```
1 #include <stdio.h>
2 int e_gcd(int A,int B) {
3     int T;
4     int R;
5     if (B>A) {
6         T = A; A = B; B=T;
7     }
8     while (B>0) {
9         R = A%B;
10        A=B; B=R;
11    }
12    return(A);
13 }
14
15 int main() {
16     printf("%d\n",e_gcd(128,24));
17     printf("%d\n",e_gcd(2147483650,6));
18 }
```

このプログラムにおいて $A\%B$ の代わりに $\text{srem}(A,B)$ を, $>$ の代わりに $\text{deg}(A)$ (A の次数) を使い, 小修正すれば多項式 gcd も求まる.

コード 4: gcd 計算 asir 版

```
1 def e_gcd(A,B) {
2     if (B>A) {
3         T = A; A = B; B=T;
4     }
5     while (B>0) {
6         R = A%B;
7         A=B; B=R;
8     }
9     return(A);
10 }
11
12 e_gcd(128,24);
13 e_gcd(2^31+2,6);
14 /* 2^31+2=2147483650 */
```
