

## C 言語について

FORTRANがFormula Translationの略であるように数式をそのまま記述してプログラムとして扱えるようにしたものに対してC言語はシステム(UNIXなどのOS)を記述できるように設計されたものなのでFORTRANに対して文法規則も多く、複雑な処理をすることが可能である。ここでは、その一部のみについて解説する。

### 簡単なプログラム

例題17にあるシンプソンの公式のプログラムをCで書いたものである。

<pre>#include &lt;stdio.h&gt; double f(double x); main() {     int i,n;     double x,a,b,h,s,s1,s2;     a=0.0;     b=1.0;     scanf("%d",&amp;n);     h=(b-a)/(double)n;     s1=0.0;     for(i=2;i&lt;n-1;i+=2){         s1+=f(a+i*h);     }     s2=0.0;     for(i=1;i&lt;n;i+=2){         s2+=f(a+i*h);     }     s=(f(a)+2.0*s1+4.0*s2+f(b))*h/3.0;     printf("S=%20.17f¥n",s); } double f(double x) {     return 4.0/(1.0+x*x); }</pre>	<p>入出力ライブラリの読み込み 関数のプロトタイプ宣言 ここから主プログラム プログラム単位は{}で囲む 変数は全て宣言する必要がある 整数 倍精度実数 行の終わりはセミコロんで</p> <p>入力文、%dは整数&amp;は変数のポインタを示す (double)を入れることにより実数に変換</p> <p>2からn-2までiを2ずつ増やして繰り返す i+=2はi=i+2の意味</p> <p>数式はほぼそのままです 出力文 %20.17fは20桁で小数点以下17桁 ¥nは改行 関数の定義</p> <p>戻り値はこのように書く</p>
---	--

コンパイルは

```
% gcc ex17.c -lm
```

(-lmは数学ライブラリを使うとき必要)

その他に

```
if() then { } else { }
```

```
while(){ }
```

など

配列の例(例題21 連立一次方程式)

<pre> #include &lt;stdio.h&gt; void leq(double a[][11], double b[],double x[], int n); main() {     int i,j,n;     double a[11][11],b[11],x[11];     scanf("%d",&amp;n);     for(i=1;i&lt;n+1;i++){         for(j=1;j&lt;n+1;j++){             scanf("%lf",&amp;a[i][j]);         }         scanf("%lf",&amp;b[i]);     }     printf("Problem¥n");     for(i=1;i&lt;n+1;i++){         for(j=1;j&lt;n+1;j++){             printf("%6.3f ",a[i][j]);         }         printf("   %6.3f¥n",b[i]);     }     leq(a,b,x,n);     printf("Solution¥n");     for(i=1;i&lt;n+1;i++){         printf("%10.7f¥n",x[i]);     } } void leq(double a[][11], double b[],double x[], int n) {     int i,j,k;     double p,q,s;     for(k=1;k&lt;n;k++){         p=a[k][k];         for(j=k+1;j&lt;n+1;j++){             a[k][j]=a[k][j]/p;         }         b[k]=b[k]/p;         for(i=k+1;i&lt;n+1;i++){             q=a[i][k];             for(j=k+1;j&lt;n+1;j++){                 a[i][j]-=q*a[k][j];             }             b[i]-=q*b[k];         }     }     x[n]=b[n]/a[n][n];     for(k=n-1;k&gt;0;k--){         s=b[k];         for(j=k+1;j&lt;n+1;j++){             s-=a[k][j]*x[j];         }         x[k]=s;     } } </pre>	<p>いつも必要です。 プロトタイプ宣言</p> <p>配列の宣言です</p> <p>J++で j=j+1 %lf は倍精度に読み込む</p> <p>関数の呼び出し</p> <p>2次元配列の受け渡し方に 注意。 配列の寸法が必要</p> <p>a-=b は a=a-b</p> <p>k--は k=k-1</p>
--	---