

1 数値演算関数 211 頁

1.1 学習のポイント

sin、cos、log などの基本的な数値演算関数の使い方について学びます。
数値演算関数は、math.h の中でプロトタイプ宣言されていますので、使用に際しては、math.h をインクルードしてください。

sin(x)、cos(x)、tan(x) の引数 x の単位はラジアンです。したがって、x 度の sin の値を求めるには、

```
sin(x*3.1415927/180.0)
```

とします。

また引数はほとんどが double 型です。引数の型を誤って、

```
sin(0)
```

とすると、結果は全く保障されません。

```
sin(0.0)
```

と正しい引数の型で与えてください。

1.2 例題 46 reidai46.c

log、sqrt、exp、pow の値を計算しなさい。

```
/*
   log、sqrt、exp、pow の値を計算しなさい。
   reidai46.c
*/

#include <stdio.h>
#include <math.h>

int main()
{
    double x;

    printf("%10s%10s%10s%15s%20s\n",
           "x", "log(x)", "sqrt(x)", "exp(x)", "pow(10.0, x)");
```

```

    for(x = 1; x <= 10; x++){
        printf("%10.5f%10.5f%10.5f%15.5f%20.5f\n",
            x, log(x), sqrt(x), exp(x), pow(10.0, x));
    }

    return 0;
}

```

1.3 練習問題 46 rensyu46.c

$\sin(x)$ の値を、 x を 0.0 ~ 360.0 まで 20.0 きざみで表示しなさい。

```

/*
練習問題 46
sin(x) の値を、x を 0.0 ~ 360.0 まで 20.0 きざみで表示しなさい。
rensyu46.c
*/

#include <stdio.h>
#include <math.h>

int main()
{
    double x;
    double rd = 3.14;

    printf("    x        sin(x)\n");
    for(x = 0; x <= 360; x = x + 20.0){
        printf("%5.1f  %10.6f\n", x, sin(x * rd / 180));
    }

    return 0;
}

```

2 AIによる大規模データ処理入門

2.1 言語処理アルゴリズム

自然言語で記述された大規模データから、その特徴を抽出する方法を考えます。人工知能の歴史においては、自然言語処理の研究は、機会翻訳や自動要約などへの応用を念頭に置いて、さまざまな側面から取り組まれてきました。

2.2 n-gram による特徴抽出

自然言語で表現された文書の表層的な特徴を抽出することで、文書の特徴を表現します。

記号列の特徴を調べるために、同じような部分記号列を n 個の記号の並びとして切り出し、同じ n 個の記号の並びが繰り返し出現しているかどうかを数え上げます。このとき、 n 個の記号の並びを n -gram と呼びます。

例では、英文の入力に対して、 $n=5$ として 5-gram を作成していきます。例のように、5文字の記号の並びを先頭から作成していき、その頻度を調べます。

解析対象文

```
Alice was beginning to (5文字の記号の並び(5-gram)を  
解析対象文の先頭から作成していく。)
```

```
Alice  
lice  
ice w  
ce wa  
e was  
was
```

2.3 n-gram を作成するプログラム ngram.c

```
/*  
*****/  
/*      n-gram の作成      */  
/*      ngram.c           */
```

```

/*****/

#include <stdio.h>
#define N 5 /*n-gram の長さ n*/

/*関数のプロトタイプの宣言 */
void initngram(char ngram[]) ;
        /*n-gram の初期化*/
void putngram(char chr,char ngram[]) ;
        /* n-gram の出力      */

/*****/
/*    main() 関数    */
/*****/
int main()
{
    char chr  ;/*入力文字*/
    char ngram[N+1];/*n-gram*/

    /*n-gram の初期化*/
    initngram(ngram) ;

    /*n-gram の出力*/
    while((chr=getchar())!=EOF){
        putngram(chr,ngram) ;/*出力*/
    }

    return 0 ;
}
/*****/
/* putngram() 関数    */
/* n-gram の出力      */
/*****/
void putngram(char chr,char ngram[])
{
    int i ;

    for(i=0;i<N-1;++i)
        ngram[i]=ngram[i+1] ;
    ngram[N-1]=chr ;
}

```

```

printf("%s\n",ngram) ;
}

/*****/
/* initngram() 関数 */
/* n-gram の初期化 */
/*****/
void initngram(char ngram[])
{
    int i ;

    for(i=0;i<N;++i)
        ngram[i]=' ' ;

    ngram[N]='\0' ;/*文字列の終わり*/
}

```

2.4 ngram.exe の使い方

```
./ngram < alice.txt
```