

目次

- Q.曲線 (リサーチ) を描く
 - Q.入力した文章の単語数と各単語の文字数を表示
 - Q.配列を使ってリングバッファを実装
 - Q.平均値と標準偏差を求めよ
 - Q.四則演算機の作成
 - Q.マスターマインド(ヒット・アンド・ブロー 数当てゲーム)
 - Q.商品ごとの売り上げ個数と月ごとの売り上げ個数を集計する
 - Q.学生番号と氏名をメンバとする構造体を定義し、読み込む
 - Q.身長、体重、肥満度のメンバを加えた構造体を新たに定義し、肥満度を計算する
 - カレンダー
 - ハッシュ
 - スタック
 - 最小公倍数
 - 二進数で出力
 - ASCII文字表
 - Simpson法を使った積分
 - 指数関数exp(x)
-

Q.曲線 (リサーチ) を描く

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/14>

いろいろな整数地a,bに対して、tを媒介変数として、座標が

$x = \cos(2 \quad at)$

$y = \sin(2 \quad bt)$

で表される曲線 (リサーチ) を描く lisajous関数を作成しなさい

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/47>

```
#include <stdio.h>
void lisajous(int a,int b){
    char str[1024];
    sprintf(str,
        "echo \ "set parametric;"
        "set samples 400;"
        "plot cos(2*pi*%d*t),sin(2*pi*%d*t) \ "
        "| gnuplot -persist",a,b);
    system(str);
}
```

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/82>

```
#include <stdio.h>
#include <math.h>
#define PI 3 // およそ3
main()
{
    /* a,bの値 */
    int a = 10, b = 20, i;
    double x, y, t = 0.0;
    char c;

    /* x,yのどちらを表示するか */
```

```

printf("x or y ? -> ");
scanf("%c", &c);

/* 計算して表示 */
printf(" \n      -1          0          1 \n");
printf("t    |-----|-----| \n");

while (t < 10.0) {
    x = cos(2 * PI * a * t);
    y = sin(2 * PI * b * t);

    printf("%.2lf ", t);
    if (c == 'x')
        for (i = 0; i < (int)(10.0 * x + 10); i++)
            putc(' ', stdout);
    else if (c == 'y')
        for (i = 0; i < (int)(10.0 * y + 10); i++)
            putc(' ', stdout);
    printf("* \n");
    t += 0.1;
}

return 0;
}

```

Q.入力した文章の単語数と各単語の文字数を表示

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/15>

実行結果が次のようになるプログラムをつくりなさい。

例：

```
Please enter a statemant: This is a programming class
```

```
Total 5 words. Each word contains;
```

```
This: 4 letters.
```

```
is: 2 letters.
```

```
a: 1 letters.
```

```
programming: 11 letters.
```

```
class: 5 letters.
```

ただし、単語数の上限は30、一単語の最大文字数は20と仮定する。

文章入力にはgets関数を使用する。

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/99>

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>
#define TANGOSU 30
#define MOJISU 20
int main(void)
{
    int i;
    int tango_index;
    int count;
    char buffer[TANGOSU * MOJISU+1];
    char *splitted[TANGOSU];
    char* p;

    printf("Please enter a statement: ");

```

```

gets(buffer);

for(tango_index = 0, p=buffer, count=0; *p != 0; p++)
{
    if(!isspace(*p) && '\0' != *p)
    {
        if(count == 0)
        {
            splitted[tango_index] =p;
            tango_index ++;
        }
        count ++;
    } else
    {
        count = 0;
        if(isspace(*p))
        {
            *p = 0;
            continue;
        }
    }
}

printf("Total %d words. Each words contains; \n", tango_index);

for (i = 0; i < tango_index; ++i)
{
    printf("%s: %d letters. \n", splitted[i], strlen(splitted[i]));
}

return 0;
}

```

Q.配列を使ってリングバッファを実装

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/25>

リングバッファ

配列を使ってリングバッファを実装し、
以下の操作を行った結果取り出された値を画面に出力しなさい。

```

1 を追加
2 を追加
3 を追加
4 を追加 (*)
取り出し
取り出し
5 を追加
6 を追加 (*)
7 を追加
取り出し
8 を追加
取り出し (*)
9 を追加
10 を追加
取り出し
取り出し (*)
取り出し
取り出し
取り出し
取り出し (*)

```

(終了)

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/27>

```
#include<stdio.h>
#define QMAX 32
#define QO() printf("%d\n",qo())
int WP = 0;
int RP = 0;
int QUEUE[QMAX];
void qi(int a){
    QUEUE[WP]=a;
    WP=(WP+1)%QMAX;
}
int qo(){
    int ret=QUEUE[RP];
    RP=(RP+1)%QMAX;
    return ret;
}
int main(void){
    qi(1);qi(2);qi(3);qi(4);
    QO();QO();
    qi(5);qi(6);qi(7);
    QO();
    qi(8);
    QO();
    qi(9);qi(10);
    QO();QO();QO();QO();QO();
    return 0;
}
```

Q.平均値と標準偏差を求めよ

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/31>

ある飛行機の水平尾翼の板圧をマイクロメータで測定したデータが12個ある。
このデータをキーボードから入力して、平均板圧と標準偏差を求めるプログラムを作成せよ
[測定データ]

1.217 1.214 1.218 1.216 1.215 1.217

1.216 1.215 1.216 1.219 1.215 1.217

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/33>

```
#include <stdio.h>
#include <math.h>
#define VAL_MAX (12)
int main( void ) {
    double a[VAL_MAX], ave = 0.0, ss = 0.0, var, std;
    int i;
    for( i = 0; i < VAL_MAX; i++ ) {
        printf( "a[%d]:", i + 1 ); scanf( "%lf", a + i );
    }
    for( i = 0; i < VAL_MAX; i++ )
        ave += a[i];
    ave /= (double)VAL_MAX;
    for( i = 0; i < VAL_MAX; i++ )
        ss += ( a[i] - ave ) * ( a[i] - ave );
    var = ss / (double)VAL_MAX;
    std = sqrt( var );
    printf( "Ave = %lf, Var = %lf, Std = %lf\n", ave, var, std );
    return 0;
}
```

```
}
```

Q.四則演算機の作成

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/92>

コマンドライン版 加減乗除 演算機の作成

使い方: calc 数値1 [+|-*|/] 数値2

例:

```
calc 4 + 2 <enter>
```

```
6.00
```

上記の例に示すようなコマンドライン版の超簡単な加減乗除演算機を作成してください。
加減乗除ですから、+ - * / の4つのオペレータ（加減乗除演算子）を考えるだけで十分です。
括弧の処理はなくて構いません。

括弧による優先順位や四則演算の優先順位を考慮すると格段に難しい問題になります。

したがって本課題では考慮の必要はありません。

コマンドラインから与える「数値1」と「数値2」は実際には文字列として main 関数に渡されます。

ご注意ください。

例にあるように、コマンド名calcと数値1の間には半角スペースを入れます。

同様に数値1とオペレータの間にも半角スペースを入れます。

もちろんオペレータと数値2の間にもスペースが必要です。

例では小数点以下2桁で結果を表示しています。表示方法は各自で工夫してください。

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/96>

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv) {
    double num1, num2, result;
    char op;

    if (argc == 4) {
        num1 = atof(argv[1]);
        num2 = atof(argv[3]);
        op = *argv[2];

        switch (op) {
            case '+':
                result = num1 + num2;
                break;
            case '-':
                result = num1 - num2;
                break;
            case '*':
                result = num1 * num2;
                break;
            case '/':
                result = num1 / num2;
                break;
            default:
                printf("Unknown operator. \n");
                break;
        }

        printf("%.2f \n", result);
    } else {
        printf("Usage: \n");
    }
}
```

```

printf(" \ tcalc num1 [+|-|*|/] num2 \ n");
}

return 0;
}

```

Q.マスターマインド(ヒット・アンド・ブロー 数当てゲーム)

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/115>

- 1 : 4この1桁のすうじからなる順列を乱数で生成
- 2 : 4個の数字の順列を入力
- 3 : 生成した順列と入力の順列を比較
(数と位置が合っている[hit],数はあるけど位置が違う[blow])
- 4 : ヒット数とブロー数を表示
- 5 : 順列があたるまで入力を繰り返す

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/123>

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define N 4
int main(){
    int Number[N], In[N];
    int i, j, hit, blow;
    srand(time(NULL));
    for(i=0; i<N; i++){Number[i]=rand()%10; In[i]=0;}
    do{
        printf("%d桁の数を入力!:", N);
        for(i=0; i<N; i++){if((In[i]=getchar()) ==EOF) break;
            In[i]-='0';}
        fflush(stdin); hit=0, blow=0;
        for(i=0; i<N; i++){if(Number[i]==In[i]) hit++;}
        for(i=0; i<N; i++){for(j=0; j<N; j++){if(Number[i]==In[j]) {
            blow++;
            break;
        }}}
        for(i=0; i<N; i++) printf("%d-", In[i]);
        printf("%dHit,%dBlow \ n", hit, blow-hit);
    }while(hit!=N);
    for(i=0; i<N; i++) printf("%d", Number[i]);
    puts(" \ nComplete!!");
}

```

Q.商品ごとの売り上げ個数と月ごとの売り上げ個数を集計する

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/158>

- 1 から 3 までの 3 種類の商品について、
1 月から 1 2 月までの売り上げ個数を入力すると、
商品ごとの合計売り上げ個数と月ごとの合計売り上げ個数を表示するプログラム。
<http://pc8.2ch.net/test/read.cgi/tech/1165190320/210>

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>

const int month = 12;
const char *item[] = {"1", "2", "3"};

int main()

```

```

{
  int i,j,item_count,hyo[12][3],sum_item[3]={0},sum_month[12]={0};
  char temp[16];

  item_count = sizeof(item)/sizeof(*item);

  /* データの入力 */
  for(i = 0;i < month;i++){
    for(j = 0;j < item_count;j++){
      printf("%d月の%sの売り上げ > ",i+1,item[j]);
      fgets(temp,sizeof(temp),stdin);
      hyo[i][j] = atoi(temp);
    }
  }

  /* 商品ごとの合計売り上げ計算 */
  for(i = 0;i < month;i++){
    for(j = 0;j < item_count;j++){
      sum_item[j] += hyo[i][j];
    }
  }

  /* 月ごとの合計売り上げ計算 */
  for(i = 0;i < month;i++){
    for(j = 0;j < item_count;j++){
      sum_month[i] += hyo[i][j];
    }
  }

  /* 表示 */
  printf(" \n \n");
  for(i = 0;i < item_count;i++){
    printf("%sの合計売り上げ : %d \n",item[i],sum_item[i]);
  }
  printf(" \n");
  for(i = 0;i < month;i++){
    printf("%d月の合計売り上げ : %d \n",i+1,sum_month[i]);
  }

  return 0;
}

```

Q.学生番号と氏名をメンバとする構造体を定義し、読み込む

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/209>

学生番号（3桁の整数）と氏名（アルファベットで姓のみ）をメンバとする構造体を定義し、キーボードから学生番号と氏名を5名分読み込むプログラムを作成しなさい。

ただし、3桁以外の整数値を入力した場合は再入力するようにし、最後に5名分のデータをすべて表示すること。

〔実行結果〕

```

学生番号（3桁の整数）> 444
氏名（アルファベットで姓のみ）> Takeda

```

```

.....
学生番号（3桁の整数）
学生番号（3桁の整数）> 99
学生番号（3桁の整数）> 631
氏名（アルファベットで姓のみ）> Sato

```

学生番号 (3桁の整数) > 107
氏名 (アルファベットで姓のみ) > Furukawa

[5名分のデータ表示例]

学生番号 : 444 氏名 : Takeda
学生番号 : 221 氏名 : Kato
学生番号 : 859 氏名 : Azuma
学生番号 : 631 氏名 : Sato
学生番号 : 107 氏名 : Furukawa

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/236>

```
#include <stdio.h>
struct gakusei {
    int number;
    char name[1000];
};
main()
{
    int i = 0;
    struct gakusei g[5];
    while (i < 5) {
        printf("学生番号(3桁の整数)> ");
        scanf("%d", &g[i].number);
        if (100 <= g[i].number && g[i].number < 1000) {
            printf("氏名(アルファベットで姓のみ)> ");
            scanf("%s", g[i].name);
            i++;
        }
    }
    printf(" \n");
    while (i-- > 0)
        printf("学生番号: %d 氏名: %s \n", g[4-i].number, g[4-i].name);
    return 0;
}
```

Q.身長、体重、肥満度のメンバを加えた構造体を新たに定義し、肥満度を計算する

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/209>

の構造体に、身長、体重、肥満度のメンバ(いずれも浮動小数点数)を加えた構造体を新たに定義する。

また、あらかじめ5名分の学生番号と氏名のデータ(任意、実際の学生番号と一致する必要は全くない)で初期化しておく。

まず、この5名のデータにキーボードから身長[m]、体重[kg]のデータ(任意)を入力する。

次に、個人ごとの構造体データへのポインタを受け取り、身長、体重から肥満度[%]を計算する関数を作成し、

その関数を用いて各人の肥満度を表示するプログラムを作成しなさい。

ただし、肥満要注意の人(肥満度30%以上)には、任意の注意マークをつけること。肥満度は次の式で計算され、

0%のとき標準体重、正の値は肥満型、負の値は痩せ型を表わす(BMI (Body Mass Index)の計算式より)。

肥満度[%] = { 体重 / (身長 × 身長 × 22) } × 100 - 100

<http://pc8.2ch.net/test/read.cgi/tech/1165190320/236>

```
#include <stdio.h>
struct gakusei {
    int n;
    char a[1000];
    double s, t;
```

```

};
main()
{
    void print(struct gakusei);
    int i = 0;
    struct gakusei g[5] = { {444, "A", 0, 0},
                            {445, "B", 0, 0},
                            {446, "C", 0, 0},
                            {447, "D", 0, 0},
                            {448, "E", 0, 0} };
    while(i<5) {
        printf("学生番号 %d: 身長[m]> ", g[i].n);
        scanf("%lf", &g[i].s);
        printf("学生番号 %d: 体重[kg]> ", g[i].n);
        scanf("%lf", &g[i].t);
        i++;
    }
    printf(" \n");
    while(i-->0)
        print(g[4-i]);
    return 0;
}
double bmi(struct gakusei *g)
{
    return g->t/(g->s * g->s * 22)*100-100;
}
void print(struct gakusei g)
{
    int i;
    double bmi(struct gakusei *);
    char *chr(char *, char *);
    char s[1000], a[50], *b;
    printf("学生番号: %d 氏名: %s BMI: %lf", g.n, g.a, bmi(&g));
    if(bmi(&g)>30){
        switch (i = abs((((int)g.a[0]+g.n)%9*(g.n%13+(int)(bmi(&g)*10))%3)) {
            case 0 :
                strcpy(s, " ( ^ ^ ) ピザキメェおwww ");
                b = chr(s, a);
                memcpy(b, "vip? no vip?", 12);
                puts(s);
                break;
            case 1 :
                strcpy(s, " ( ^ ^ ) アウアウ!! ");
                b = chr(s, a);
                memcpy(b, "vip? no vip?", 12);
                puts("");
                puts(s);
                puts(" ≡ ");
                puts(" / \ ");
                break;
            case 2 :
                strcpy(s, " ( ^ ^ * ) おっおっおwww ");
                b = chr(s, a);
                memcpy(b, "vip? no vip?", 12);
                puts(s);
                break;
            default :

```

```

        break;
    }
}
else
    puts("");
}
char *chr(char *s, char *a)
{
    struct gakusei g;
    strcpy(g.a, s);
    if(strncmp(g.a, "gakusei", 2)==0){
        if(g.a[strlen(g.a)-1]==0x20)
            return a;
        else
            return NULL;
    }
    else
        return NULL;
}

```

カレンダー

月のカレンダーを出力する

```

#include<stdio.h>
int main(){
    int days[] = {31,28,31,30,31,30,31,30,31,31,30,31,30,31};
    int y, m, d, i, k;

    printf("y=");
    scanf("%d", &y);
    printf("m=");
    scanf("%d", &m);
    if(1583>y || y>4000 || 1>m || m>12) return 1;
    printf(" \n%d/%d \n \n", y, m);

    d = days[m-1];
    if(m==2 && (y%4==0&& y%100!=0 || y%400==0)) d++;

    if(m==1 || m==2){
        y--;
        m += 12;
    }
    k = (y+(int)(y/4)-(int)(y/100)+(int)(y/400)+(int)((13*m+8)/5)+1) % 7;

    printf("日 月 火 水 木 金 土 \n");
    for(i=0; i<k; i++) printf("   ");
    for(i=1; i<=d; i++){
        printf("%2d ", i);
        if(++k%7==0) printf(" \n");
    }
    printf(" \n");
    return 0;
}

```

ハッシュ

<http://pc10.2ch.net/test/read.cgi/tech/1166969820/684>

引数として与えられた文字列から、

ハッシュ値を計算するハッシュ関数を作成し、
以下の文字列についてハッシュ値を計算して
画面に出力しなさい。

ハッシュ値を計算する文字列

"Matsuzaka"

"Igawa"

"Kuwata"

"Okajima"

ハッシュ関数の内容

与えられた文字列の、一つ一つの文字の文字コードを合計し、
最後にハッシュ表の大きさを割った余りを計算する。

ハッシュ表の大きさは7とする

```
#include<stdio.h>
#define HASH_SIZE 7
int hash(char* s){
    int h = 0;
    while(*s != '\0') h += *s++;
    return h % HASH_SIZE;
}
int main(){
    printf("Matsuzaka %d \n",hash("Matsuzaka"));
    printf("Igawa %d \n",hash("Igawa"));
    printf("Kuwata %d \n",hash("Kuwata"));
    printf("Okajima %d \n",hash("Okajima"));
    return 0;
}
```

スタック

<http://pc10.2ch.net/test/read.cgi/tech/1168663908/941>

```
#include<stdio.h>
#define N 5
int push(int data, int *sp, int *STK){
    if(*sp <= 0) return 0;
    STK[--(*sp)] = data;
    return 1;
}
int pop(int *dataP, int *sp, int *STK){
    if(*sp >= N) return 0;
    *dataP = STK[(*sp)++];
    return 1;
}
void dsp_stack(int top, int *a){
    while(top++ < N)
        printf("%d: %d \n",top-1, a[top-1]);
}
int main(){
    int data,stk[N],top=N;
    char cmd;
    while(scanf("%c",&cmd) != EOF){
        if (cmd=='i'){
            scanf("%d", &data);
            if(push(data,&top,stk)==1)
                dsp_stack(top,stk);
        }else if(cmd=='o'){
            if(pop(&data,&top,stk)==1){

```

```
        printf("popped data= %d \n",data);
        dsp_stack(top,stk);
    }
}
return 0;
}
```

最小公倍数

```
#include<stdio.h>
int main(){
    int a, b, i = 0;
    scanf("%d %d", &a, &b);
    while(b&&(i+=a))
        if(i%b == 0) break;
    printf("lcm(%d,%d) = %d \n", a, b, i);
    return 0;
}
```

二進数で出力

```
#include<stdio.h>
void BinDisp(int n){
    if(n > 1) BinDisp(n / 2);
    printf("%d", n % 2);
}
int main(){
    BinDisp(10);
    printf(" \n");
}
```

ASCII文字表

```
#include<stdio.h>
int main(){
    int i, j, k;
    for(i=0; i<=0xF; i++){
        for(j=0; j<=7; j++){
            k = i*8 + j;
            if(0x20<=k && k<=0x7e)
                printf(" %c ", k);
            else
                printf("** ");
        }
        printf(" \n");
    }
    return 0;
}
```

Simpson法を使った積分

シンプソンの公式を用いて定積分を求めるプログラムを作成し、以下2つの定積分を計算せよ。ただし刻み幅 $h=0.1$ とする。

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

```

typedef double (*SingleVFunc)(double);
double quadInterpol( SingleVFunc fun, double begin, double h )
{
    return (fun( begin ) + 4.0*fun( begin + h ) + fun( begin + 2.0*h )) * h/3.0;
}

double integral( SingleVFunc fun, double begin, double end, double h )
{
    double result = 0.0;
    double pos = begin;
    do
    {
        result += quadInterpol( fun, pos, h );
        pos += 2.0*h;
    } while( pos < end );
    return result;
}

double f(double x) { return exp(-x*x); }
double g(double x) { return 1.0/sqrt(1 + x*x*x*x); }

int main(void)
{
    const double diff = 0.1;
    printf("result=%f \n", integral(f,0.0,1.0,diff));
    printf("result=%f \n", integral(g,0.0,1.0,diff));
    return 0;
}

```

指数関数exp(x)

```

#include <stdio.h>

double myexp(double x)
{
    double coefficient = 1.0;
    double value = 0.0;
    int k = 1;
    while( coefficient >= 1.0e-6 )
    {
        value += coefficient;
        coefficient *= x / (double)k;
        ++k;
    }
    return value;
}

int main(void)
{
    double x;
    printf(" \t x \t myexp(x) \n");
    for( x=1.0; x<=10.0; ++x )
        printf(" \t%.1f \t%13.5f \n",x,myexp(x));
    return 0;
}

```