データの配列の基本（数学購読）

５月８日課題（６月６日まで）

課題１ 二つのベクトルの外積を計算し表示するプログラムを作れ.

課題２ 行列の転置行列を計算し表示するプログラムを作れ.

課題３ 二つの行列の積を計算し表示するプログラムを作れ.

課題４ 与えられた文字列の順序を逆順に表示するプログラムを作れ.

課題５ printf 関数を使わずに与えられた整数を１６進表示するプログラムを作れ.

実習で利用するサンプルプログラム

prog3-1.c （配列の基本）

```
#include <stdio.h>
int main( int argc, char *argv[] )
{
  int i,n=10;
  /* 整数型のデータ配列を１０個作成する */
  int a[10];
  /*  配列は必ず初期化しなければならない. */
  /* 初期化しないとどんなデータが入っているかわからないことを見るため何もせず表示. */
  printf("Without initialization\n");
  for( i=0;i<n;i++ ){
    printf("a[%d]=%d\n", i, a[i]);
```

```
  }
  /* 初期化する */
  for( i=0;i<n;i++ ){
    a[i] = i%3;   /* %演算はmod演算と同じ. */
  }
  printf("---------------\n");
  /* データの出力 */
  printf("With initialization\n");
  for( i=0;i<n;i++ ){
    printf("a[%d]=%d\n", i, a[i]);
  }
  exit(0);
}
```

prog3-2.c （ベクトル演算の例）

```
#include <stdio.h>
int main( int argc, char *argv[] )
{
  int i;
  double r,alpha;
  /* ４次元ベクトルをdouble実数の配列として宣言.  */
  /* 宣言と同時に以下のように初期化することができる */
  double a[4]={1.0, 2.0,3.0,4.0};
  double b[4]={2.0,-4.0,1.0,-3.0};
  /* c,d,eは計算結果を代入するので初期化はしない */
  double c[4],d[4],e[4];
  /* ベクトルの足し算 */
  for(i=0;i<4;i++){
    c[i] = a[i]+b[i];
  }
  /* ベクトルの引き算 */
  for(i=0;i<4;i++){
    d[i] = a[i]-b[i];
  }
  /* ベクトルのスカラー積 */
  alpha = 0.1;
  for(i=0;i<4;i++){
```

```
    e[i] = alpha*a[i];
  }
  /* ベクトルの内積 */
  r = 0.0;
  for(i=0;i<4;i++){
    r += a[i]*b[i];
  }
  /* 結果の表示 */
  printf("a = ( %lf,%lf,%lf,%lf )\n", a[0], a[1], a[2], a[3] );
  printf("b = ( %lf,%lf,%lf,%lf )\n", b[0], b[1], b[2], b[3] );

  printf("a+b = ( ");
  for(i=0;i<4;i++){
    printf("%lf ", c[i] );
  }
  printf(")\n");
  printf("a-b = ( ");
  for(i=0;i<4;i++){
    printf("%lf ", d[i] );
  }
  printf(")\n");
  printf("%lf*a = ( ", alpha);
  for(i=0;i<4;i++){
    printf("%lf ", e[i] );
  }
  printf(")\n");
  printf("(a,b) = %lf\n", r );
  exit(0);
}
```

prog3-3.c （行列演算の例）

```
#include <stdio.h>
int main( int argc, char *argv[] )
{
  int i,j;
  /*  ４×４の行列を double 実数の二重配列として宣言  */
  double a[4][4],b[4][4],c[4][4],d[4][4],e[4][4];
```

```
double va[4]={1,0,-2,1},vb[4],ma[16];
/* 初期化 */
for(i=0;i<4;i++){
  vb[i] = 0.0;
  for(j=0;j<4;j++){
    a[i][j] = (double)i*(double)j;
    b[i][j] = (double)i/(double)(j+1);
    c[i][j] = d[i][j] = e[i][j]= 0;
  }
}
/* 行列の和と差の計算 */
for(i=0;i<4;i++){
  for(j=0;j<4;j++){
    c[i][j] = a[i][j]+b[i][j];
    d[i][j] = a[i][j]-b[i][j];
  }
}
/* 行列 a とベクトル va の積の計算  */
for(i=0;i<4;i++){
  for(j=0;j<4;j++){
    vb[i] += a[i][j]*va[j];
  }
}
/* 二重配列は通常の配列に置き換えることもできる */
for(i=0;i<4;i++){
  for(j=0;j<4;j++){
    ma[i*4+j] = a[i][j];
  }
}
/* 結果の表示 */
printf( "a = \n" );
for(i=0;i<4;i++){
  printf( " %lf %lf %lf %lf\n", a[i][0], a[i][1], a[i][2], a[i][3] );
}
printf( "b = \n" );
for(i=0;i<4;i++){
  printf( " %lf %lf %lf %lf\n", b[i][0], b[i][1], b[i][2], b[i][3] );
}
```

```
  printf( "a+b = \n" );
  for(i=0;i<4;i++){
    for(j=0;j<4;j++){
      printf( " %lf", c[i][j] );
    }
    printf( "\n" );
  }
  printf( "a-b = \n" );
  for(i=0;i<4;i++){
    for(j=0;j<4;j++){
      printf( " %lf", d[i][j] );
    }
    printf( "\n" );
  }
  printf("va = ( ");
  for(i=0;i<4;i++){
    printf( "%lf ", va[i] );
  }
  printf( ")\n" );
  printf("a*va = ( ");
  for(i=0;i<4;i++){
    printf( "%lf ", vb[i] );
  }
  printf( ")\n" );
  printf( "a = \n" );
  for(i=0;i<4;i++){
    for(j=0;j<4;j++){
      printf( "%lf ", ma[i*4+j] );
    }
    printf("\n");
  }
  exit(0);
}
```

prog3-4.c (char 型の配列＝ ”文字列 ”の例)

```
#include <stdio.h>
int main( int argc, char *argv[] )
```

```
{
  int i;
  /* char 型の変数は英文字を一つ格納するのに用いる */
  /* char 型の配列は文字の集まり＝「文字列」として用いることができ
る */
  /* 以下の例では３０文字分の配列を宣言している */
  char a[30];
  /* 文字列の初期化も通常の配列の初期化と同様にできる */
  char b[]="This is a pen.";
  /* 文字列の初期化は通常の配列と同様に一つ一つ文字を代入することで
も可能である */
  a[0] = 'I';  a[1] = ' ';  a[2] = 'a';  a[3] = 'm';  a[4] = ' ';
  a[5] = 'a';  a[6] = ' ';  a[7] = 'm';  a[8] = 'a';  a[9] = 't';
  a[10] = 'h';  a[11] = ' ';  a[12] = 's';  a[13] = 't';  a[14] = 'u';
  a[15] = 'd';  a[16] = 'e';  a[17] = 'n';  a[18] = 't';  a[19] = '.';
  /* まず char 型の英文字を一字表示するには %c を使えばよい */
  printf( "%c\n", a[12] );
  /* char 型の文字列の表示には%s を使う */
  printf( "%s\n", b );
  /* 単なる char 型の配列と文字列の違いは，配列の最後のデータが  */
  /* char 配列の最後のデータが文字列の終わりを表す  \0 であること */
  /* 上の a の初期化では\0 がないので，文字列として不明確である．*/
  printf( "%s\n", a );
  /*きちんと文字列の最後として\0 を代入して表示 */
  a[20] = '\0';
  printf( "%s\n", a );
  exit(0);
}
```