**課題１**

ファイル名：montecarlo.c

目的 ：モンテカルロ法による円周率の計算

プログラムをint pi(int n, double \*pi)という関数とmain関数に分けられる。

関数piで円周率を推定し、推定値が \*piに代入される。piを推定するために、n回乱数(x, y)をとり、の条件を満たすものを数え、その数をmとする。で求める。

int pi(int n, double \*pi){

int i; //for文のため

int m = 0;

time\_t t;

//プログラムをrunするごとに、違う結果が出るように、以下の文が必要

srand((unsigned) time(&t));

for (i = 0; i < n; i++){

//乱数を1-1000の間からとる

int x = rand() % 1000 + 1;

int y = rand() % 1000 + 1;

if ((x\*x)+(y\*y) < 1000000){

//n個の乱数(x,y)のなか、以上の条件を満たすものを数える。

m = m+1;

}

}

if (m > 0){

\*pi = 4\*((double)m)/n; //求めたpiの値を\*piに代入する

return 0;

} else {

return -1;

}

}

main関数でnを決めて（大きいほど、真のpi値に近い）、初期値0のdouble \*bを宣言する。

int main(){

int n = 100000;

double \*b;

printf("initial: %f\n", **b);** //\*bの初期値を表示する

pi(n, b); //piを実行

printf("result: %f\n", \*b); //pi実行後の\*bを表示

return 0;

}

**課題２**

ファイル名：sort.c

目的 ：intの配列をソートする

プログラムがvoid sort\_int\_array(int \*ary, int ary\_size)とmain関数からなる。

関数sort\_int\_arrayはバブルソートで配列をソートする。i = 0から

void sort\_int\_array(int \*ary, int ary\_size){

int i, j;

int temp;

for (i = 0; i < ary\_size; i++){

int k = ary\_size - 1; //配列の末尾を比較する必要がないの　　 で、-1にする

for (j = 0; j < k; j++){

if (ary[j+1] < ary[j]){

//ary[j]とary[j+1]をスワップする

temp = \*(ary+j);

\*(ary+j) = \*(ary+j+1);

\*(ary+j+1) = temp;

}

}

k = k - 1;

}

}

main関数でソートしたい配列を宣言する

int main(){

int ary[] = {1, 6, 7, 8, 2, 3, 2, 1, 5, 7, 2, 13, 253, 2, 2, 3, -3, -233, 934, 2, 334, 97};

int ary\_size = 22;

sort\_int\_array(ary, ary\_size);

printf("array: ");

for (int i = 0; i < ary\_size; i++){

printf("%d ", \*(ary+i));

}

printf("\n");

return 0;

}

**課題３**

ファイル名：