

前期末試験 (2E 情報工学概論)

2005 年 9 月 26 日

1 基本 C 言語プログラム

[問 1] プログラムの動作とそのソースが示されている。[ア]~[ノ]に入る適当な文を与えられた選択肢から選べ。回答欄には、a~d の記号を記入すること。

1.1 1 行表示

- 以下の 1 文を表示する。

Hello World !!

```
#include <stdio.h>
int main(){
    
    return 0;
}
```

選択肢

- の選択肢
- (a) scanf("Hello World !!\n");
- (b) write("Hello World !!\n");
- (c) put("Hello World !!\n");
- (d) printf("Hello World !!\n");

1.2 三角関数の値の表示

- キーボードから角度 [deg] を読み込む。なお、角度は倍精度実数とする。
- 読み込んだ角度の三角関数の値を表示する。

```
#include <stdio.h>


int main(){

    double rad, deg;

    /* --- 入力のメッセージと整数入力--- */
    printf("角度 [deg]? \t");
    scanf("%lf", &deg);

    /* --- 計算と表示 --- */
    rad = deg / 180.0 * 3.14159265;
    printf("sin = %f\n", sin(rad));
    printf("cos = %f\n", cos(rad));
    printf("tan = %f\n", tan(rad));

    return 0;
}
```

選択肢

- の選択肢
- (a) #include <c.h>
- (b) #include <c_language.h>
- (c) #include <mathematics.h>
- (d) #include <math.h>

1.3 整数の商と余り

- キーボードから2つの整数を読み込む。
- 読み込んだ2つの整数の商と余りを計算し、表示する。

```
#include <stdio.h>

int main(){
    

    /* --- 入力のメッセージと整数入力--- */
    printf("最初の整数?\t");
    scanf("%d",&i);
    printf("次の整数?\t");
    scanf("%d",&j);

    /* --- 計算と表示 ---*/
    printf("%d/%d=\n",i,j);
    
    

    return 0;
}
```

選択肢

- の選択肢
 - (a) double i,j;
 - (b) int i,j;
 - (c) complex i,j
 - (d) int x, y
- の選択肢
 - (a) printf("\t商\t%d\n",i/j);
 - (b) printf("\t商\t%d\n",i÷j);
 - (c) printf("\t商\t%d\n", $\frac{i}{j}$);
 - (d) printf("\t商\t%d\n",quotient(i,j));
- の選択肢
 - (a) printf("\t余り\t%d\n",i+j);
 - (b) printf("\t余り\t%d\n",i*j);
 - (c) printf("\t余り\t%d\n",i#j);
 - (d) printf("\t余り\t%d\n",i%j);

1.4 度からラジアンへの変換

- キーボードから角度 [deg] を読み込む。なお、角度は倍精度実数とする。
- 読み込んだ角度の単位をラジアン [rad] に変換し、表示する。

```
#include <stdio.h>

int main(){
    

    /* --- 入力のメッセージと整数入力--- */
    printf("角度 [deg]?\t");
    

    /* --- 計算と表示 ---*/
    
    printf("%f [rad]\n",rad);

    return 0;
}
```

選択肢

- の選択肢
 - (a) double rad, deg;
 - (b) double x, y, z;
 - (c) int rad, deg;
 - (d) angle rad, deg
- の選択肢
 - (a) scanf("%lf",°);
 - (b) scanf("%lf",deg);
 - (c) printf("%lf",°);
 - (d) read("%lf",°);
- の選択肢
 - (a) deg/180.0*3.14159265=rad;
 - (b) rad <- deg/180.0*3.14159265
 - (c) rad=deg÷180.0*3.14159265;
 - (d) rad=deg/180.0*3.14159265;

1.5 whileを使ったループ

- while 文を使って、1~1000 までの整数の合計を計算する。

```
#include <stdio.h>

int main(){

    int i, sum;

    /*--- 初期化 ---*/
    sum=0;
    i=1;

    ケコサ    /* sum=sum+i */

}

printf("sum = %d\n",sum);

return 0;
}
```

選択肢

- ケ の選択肢
 - (a) while_do(i<=1000)
 - (b) loop(i<=1000)
 - (c) while(1000)
 - (d) while(i<=1000){
- コ の選択肢
 - (a) sum+i=sum
 - (b) sum=i
 - (c) sum+=i;
 - (d) i=j
- サ の選択肢
 - (a) i++;
 - (b) i--
 - (c) i**
 - (d) &i

1.6 do~whileを使ったループ

- do~while 文を使って、1~1000 までの整数の合計を計算する。

```
#include <stdio.h>

int main(){

    int i, sum;

    /*--- 初期化 ---*/
    sum=0;
    i=1;

    シ

    sum+=i;                                /* sum=sum+i */
    i++;

    ス

    printf("sum = %d\n",sum);

    return 0;
}
```

選択肢

- シ の選択肢
 - (a) while{
 - (b) do{
 - (c) done{
 - (d) does{
- ス の選択肢
 - (a) }while(i<=1000);
 - (b) }if(i<=1000);
 - (c) else if(i<=1000);
 - (d) continue(i<1000);

1.7 配列の利用

- 10 個の整数をキーボードから読み込み、配列に格納する。
- 読み込んだ整数の偶数と奇数の合計を計算する。

```
#include <stdio.h>
#define N 10

int main(){
    char temp;
    
    int sum_gusu, sum_kisu;

    /* キーボードから読み込んだ整数を配列へ */
    for(i=0; i<N; i++){
        printf("%d 番目の整数?\t", i+1);
        
    }

    /* --- 合計を入れる変数を 0 に初期化 --- */
    sum_gusu = 0;
    sum_kisu = 0;

    /* --- 偶数と奇数の合計の計算 --- */
    for(i=0; i<N; i++){
         /* 偶数ならば */
        sum_gusu += input_num[i];
         /* さもなければ */
        sum_kisu += input_num[i];
    }

    printf("偶数の和\t%d\n", sum_gusu);
    printf("奇数の和\t%d\n", sum_kisu);

    return 0;
}
```

選択肢

- の選択肢

- (a) double i, input_num[N]
- (b) int[10] i, input
- (c) dimension i, input_num[N]
- (d) int i, input_num[N];

- の選択肢

- (a) scanf("%d%c", &input_num[i], &temp);
- (b) scanf("%d%c", &input_num, &temp);
- (c) scanf("%d%c", input_num[i], temp);
- (d) scanf(&input_num[i], &temp);

- の選択肢

- (a) if(input_num[i]%2 == 0){
- (b) if(input_num[i]%3 == 0){
- (c) if(input_num%2 == 0){
- (d) if(input_num[i]%2 == 2){

- の選択肢

- (a) }break{
- (b) }else{
- (c) }continue{
- (d) }while{

1.8 三角関数表の作成

- ファイルに 0 ~ 360 度まで、1 度毎に、三角関数の値を書き込む。
- ファイルの各行には、角度 θ [deg] と $\sin \theta$ 、 $\cos \theta$ 、 $\tan \theta$ の値をタブ区切りで書く。

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

int main(){

    int theta;
    double rad, s, c, t;
    ツ

    /* ファイルのオープン */
    テ

    for(theta=0; theta<=360; theta++){
        /* 度からラジアンへ */
        rad=theta/180.0*3.1415926;

        /* 三角関数の計算 */
        s=sin(rad);
        c=cos(rad);
        t=tan(rad);

        /* 書き込み */
        ト
    }

    /* ファイルのクローズ */
    fclose(fp);

    return 0;
}
```

選択肢

- ツ の選択肢

- (a) DISK *fp;
- (b) TABLE *fp;
- (c) DRIVE *fp;
- (d) FILE *fp;

- テ の選択肢

- (a) fp=fopen("triangle_table.txt","w");
- (b) fopen("triangle_table.txt","w")=fp;
- (c) fp=fscanf("triangle_table.txt","w");
- (d) fp=fopen("triangle_table.txt","w");

- ト の選択肢

- (a) fprintf(fp,"%d\t%f\t%f\t%f\n",theta,s,c,t);
- (b) fprintf(fp,"%d\n%f\n%f\n%f\n",theta,s,c,t);
- (c) fwrite(fp,"%d\t%f\t%f\t%f\n",theta,s,c,t);
- (d) fwrite(fp,"%d\n%f\n%f\n%f\n",theta,s,c,t);

1.9 三角形の面積

- キーボードから、三角形の3辺の長さを読み込む。
- ヘロンの公式を用いて、三角形の面積を計算する。
- もし、入力した3辺で三角形が出来なければ、“三角形は出来ません”と表示する。2辺の合計が他の1辺より短いとき、三角形は出来ない。

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

double heron(double x, double y, double z);
/* ===== */
/*      main 関数                                */
/* ===== */
int main(){
    double a, b, c, s;
    char temp;

    printf("三角形の3辺の長さ?\n");
    scanf("%lf%lf%lf%c", &a, &b, &c, &temp);

    s=heron(a, b, c);

    if(s<-900){
        printf("三角形は出来ません\n");
    }else{
        printf("面積は、%f です。 \n",s);
    }

    return 0;
}

/* ===== */
/* ヘロンの公式より、三角形の面積を求める関数    */
/* ===== */
double heron(  ナ ) {
    double s, menseki;

    if(  二 ){
        return -999;
    }

    s=(x+y+z)/2.0;
    menseki=sqrt(s*(s-x)*(s-y)*(s-z));
     又
}
```

選択肢

- ナ の選択肢
 - (a) int x, int y, int z
 - (b) double x, double y, double z
 - (c) double a, double b, double c
 - (d) double x, y, z
- 二 の選択肢
 - (a) (x+y)=z || (y+z)=x || (z+x)=y
 - (b) (x+y)<z || (y+z)<x || (z+x)<y
 - (c) (x+y+z)>0
 - (d) sin(x)+sin(y)+sin(z)<0
- 又 の選択肢
 - (a) end;
 - (b) return menseki;
 - (c) stop menseki;
 - (d) goto main;

1.10 変数の入れ替え

- 整数型の変数 a, b, c に入っている値を、 $a \rightarrow b$ 、 $b \rightarrow c$ 、 $c \rightarrow a$ と入れ替える。
- 入れ替えには、プログラマーが作成する関数 rot() を使う。

```
#include <stdio.h>

int rot(int *x, int *y, int *z);
/* ===== */
/*      main 関数      */
/* ===== */
int main(){

    int a, b, c;

    a=1;
    b=2;
    c=3;

    printf("a=%d\tb=%d\tc=%d\n", a, b, c);

    rot(&a, &b, &c);    /* 入れ替えの関数 */

    printf("a=%d\tb=%d\tc=%d\n", a, b, c);

    return 0;
}

/* ===== */
/*      値をサイクリックに入れ替える      */
/* ===== */
int rot(  ){
int rot(){
    int temp;

    temp=*z;

    *z = *y;
    *y = *x;
    

    return 0;
}
```

選択肢

- の選択肢

- (a) int *x, int *y, int *z
- (b) int x, int y, int z
- (c) double *x, double *y, double *z
- (d) double x, double y, double z

- の選択肢

- (a) *x=*z
- (b) x=y;
- (c) temp=*x;
- (d) *x = temp;

2 文法

[問 1] プログラム中の以下のように書かれている。この動作を説明せよ。

```
#define TEST 100000
```

[問 2] プログラム中の以下のように書かれている。この動作を説明せよ。

```
while(fscanf(fp, "%d%lf%lf%lf", &theta, &s, &c, &t) != EOF){  
    printf("%d\t%f\t%f\t%f\n", theta, s, c, t);  
}
```

[問 3] プログラム中の以下のように書かれている。整数型変数 i の値を示せ。

```
i=(int)sqrt(200.0);
```

3 プログラム

[問 1] for 文を使って、1~1000 まで、計算するプログラムを作成せよ。

[問 2] 以下の要領で、円の面積から円周率を計算する。

- 図 1 の円の面積は π である。この面積を求めることにより円周率 π を計算する。
- 円の $1/4$ 円の中に複数の帯がある。これらの帯状の部分の面積の合計は円の面積の $1/4$ に近い。この帯からなる部分の面積を 4 倍することにより、円の面積を求める。
- 帯は、一定の幅で、区間 $0 \leq x \leq 1$ を n 等分する事により作成する。この n を分割数という。

分割数 n 数が 2~1000 と変化させた場合の円の面積、すなわち円周率を出力するプログラムを作成せよ。

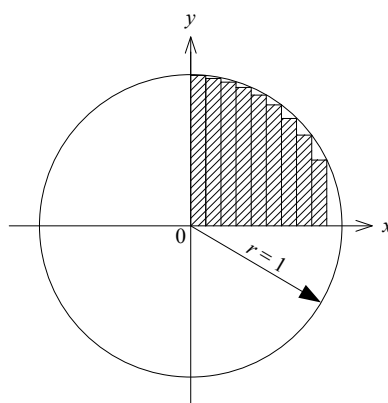


図 1: 円の面積