セメスタ課題に関する注意：(他の課題とは違って) セメスタ課題では，作成したプログラムに関する**報告書 (レポート)** も提出しなければいけません．この報告書には，実験報告書と同様に，目的，原理，考察を書かなければいけません．また，実験報告書にはあまり書くことがないかもしれませんが，作成物 (プログラム) の使用法も書かなければいけません．詳しくは,「セメスタ課題報告書の書き方」を参考にすること．

セメスタ課題 2(定積分)TS2

	木曜クラス
セメスタ課題 2 課題提示	6 月 26 日 (木)
セメスタ課題 2 提出締切	7 月 10 日 (木)
セメスタ課題 2 課題採点結果通知	7 月 17 日 (木)
セメスタ課題 2 再提出締切	7 月 24 日 (木)

TS2-1:定積分 (プログラム)

提出ファイル : Makefile, ソースファイル (integral.c), 入力ファイル (integral.in), 出力ファイル (integral.out)

与えられた関数に対して，定積分を計算するプログラムを作成せよ．積分の対象となる関数を $f(x)$，積分区間の下限を a，積分区間の上限を b とすると，求めるべき定積分値は,

$$\int_a^b f(x)dx = F(b) - F(a)$$

である．ただし，$F(x)$ は $f(x)$ の原始関数 ($\frac{d}{dx}F(x) = f(x)$ であるような関数) の一つである．

プログラムは，下の実行例を参考にし，後ろに示す要求仕様をみたすように作成すること．ただし，各実行例は，それぞれ異なるプログラムを実行した結果である．

実行例 1

```
b09b0yy@txx:~/TS2/1$./integral
関数 f(x)=4.0/(1.0+x*x) の定積分を行います.
積分区間の下限を入力してください : 0.0
積分区間の上限を入力してください : 1.0
区間 [0.0 , 1.0] をいくつの区間に分割して積分値を求めますか？
(2 以上の偶数を入力してください): 10

関数 f(x) の区間 [0.000000 , 1.000000] における積分値は
3.141592613939215
です.
b09b0yy@txx:~/TS2/1$
```

実行例 2

```
b09b0yy@txx:~/TS2/1$./integral
関数 f(x)=1.0/x の定積分を行います.
積分区間の下限を入力してください : 1.0
積分区間の上限を入力してください : 2.0
区間 [0.0 , 1.0] をいくつの区間に分割して積分値を求めますか？
(2 以上の偶数を入力してください): 20

関数 f(x) の区間 [1.000000 , 2.000000] における積分値は
0.693147374665116
です.
b09b0yy@txx:~/TS2/1$
```

実行例 3

```
b09b0yy@txx:~/TS2/1$./integral
関数 f(x)=1.0/x の定積分を行います.
積分区間の下限を入力してください : 1.0
積分区間の上限を入力してください : 2.0
区間 [0.0 , 1.0] をいくつの区間に分割して積分値を求めますか？
(2 以上の偶数を入力してください): 0

分割する区間の数は，2 以上の偶数でなくてはいけません.
b09b0yy@txx:~/TS2/1$
```

要求仕様 (TS2-1)(次ページ以降)

全体的な仕様:

- 様々な数学的関数 $f(x)$ の定積分を求める C 言語の関数 `calc_integral` を作成する．ここで，数学的関数 $f(x)$ は C 言語の関数 `target` としてソースコード中に与えられるとする．関数 `calc_integral` の仕様は後の説明を参照.
- 積分対象の数学的関数 $f(x)$ の関数値を計算する C 言語の関数 `target` をソースファイル中で定義する．関数 `target` の仕様は後の説明を参照.
- 積分対象の数学的関数 $f(x)$ の説明を表示する C 言語の関数 `description` をソースファイル中で定義する．関数 `description` の仕様は後の説明を参照.
- 積分を計算するアルゴリズムに応じて，導出される数値解の精度を制御するためのパラメータ (区間の分割数) を用意する．(実行例を参照すること．)
- ソースコード中の関数 `target` の定義部分および `description` の定義部分のみを書換えるだけで，任意の関数に対する定積分を計算できるようにプログラムを作成すること．すなわち，$f(x)$ の積分を人間が手作業で行う必要がないようにすること.
- グローバル変数は用いないこと.
- 入力は，以下を満すように標準入力から行なう.
  - 積分区間の下限 a，積分区間の上限 b，制御パラメータ n はこの順序に入力される.
  - 積分区間の下限 a と上限 b は，それぞれ任意の実数 (double 型の値) であり，上限が下限より大きい値であるとは限らない．関数 `calc_integral` の引数として指定できる積分区間の下限と上限の値には制限が存在するが，プログラム全体としては制限が存在しないことに注意すること．(関数 `calc_integral` は $a < b$ の場合のみ正しく計算できようにするが，プログラム全体では $a < b$,$a = b$,$a > b$ のいずれの場合にも正く定積分値を計算できるようにすること．そのために，$a > b$ の場合でも関数 `calc_integral` を利用できるはずである．)

    この条件での下限 a と上限 b を，C 言語の関数 `calc_integral` の関数の仮引数の条件に合うように適切に仮引数に渡す.
  - 数値解の精度を制御する制御パラメータ (区間分割数)n は，1 以上の自然数とする．用いるアルゴリズムに応じて制約を設けても良い．例えば，制御パラメータを 2 以上の偶数に限定しても良い.
- 出力は，以下を満すように標準出力へ行なう.
  - 関数 $f(x)$ の区間 $[a, b]$ における定積分

    $$\int_a^b f(x)dx = F(b) - F(a)$$

    の値を出力する.
  - 積分区間の下限 a が上限 b 以上の場合であっても，適切な定積分値を出力する.
  - 出力される定積分値は，計算精度を反映した十分高い精度を持つようにする.

積分対象の数学的関数 $f(x)$ の定積分値を計算する関数 calc_integral の仕様：

- 次のプロトタイプ宣言を持つ.

```
/*
機能説明：
関数 target により定義される積分対象関数の定積分値を求める関数

仮引数:
a:積分区間の下限 (任意の実数)
b:積分区間の上限 (b>a なる実数)

戻り値:
関数 target で指定された数学的関数の積分区間 [a,b] による
定積分値 (実数);
*/

double calc_integral(double a,double b,int n);
```

- 引数 a は積分区間の下限で，任意の実数値 (double 型の値) とする.
- 引数 b は積分区間の上限で，下限 a 以上の実数値 (double 型の値) とする．呼び出された段階で常に $a < b$ は満たされているとして良く，関数 calc_integral 内でエラー処理をする必要は無い.
- 引数 n は制御パラメータ (区間分割数) で 1 以上の自然数とする．この制御パラメータには，利用する定積分計算アルゴリズムに応じて必要な制約を設けても良い．もし制約がある場合でも，呼び出された段階で常に制約が満たされているとして良く，関数 calc_integral 内でエラー処理をする必要は無い．実行例参照.
- 戻り値は，求めるべき定積分

$$\int_a^b f(x)dx = F(b) - F(a)$$

の値であり，double 型の値を持つ.

- 関数 calc_integral 内では，標準入出力は行なってはならない．(printf 文による表示や，scanf 文による値の読み込みは，行なってはならない．)

積分対象関数 $f(x)$ の関数値を計算する関数 target の仕様：

- 次のプロトタイプ宣言を持つ．

```
/*
機能説明：
積分対象関数の関数値を求める関数

仮引数:
x:被演算項（任意の実数）

戻り値：
仮引数に与えられた実数値（実引数）に対する関数値
*/
double target(double x);
```

- 積分対象関数 $f(x)$ の関数値を適切に計算する．例えば，

$$f(x) = \frac{4}{1+x^2}$$

の場合は，

```
double target(double x)
{
  return 4.0 / (1.0 + x * x );
}
```

のように定義されていればよい．

- 引数 x は任意の実数 (double 型の値) とする．
- 引数 x に実数値 x が与えられたとき，戻り値は関数値 $f(x)$ の実数値 (double 型の値) とする．
- 関数 target 内では，標準入出力は行なってはならない．(printf 文による表示や，scanf 文による値の読み込みは，行なってはならない．)

積分対象関数 $f(x)$ の説明を表示する関数 description の仕様：

- 次のプロトタイプ宣言を持つ．

```
/*
機能説明：
積分対象関数の説明を表示する関数

仮引数:
なし（void）

戻り値:
なし（void）

副作用：
積分対称関数の説明を標準出力へ出力する
*/

void description();
```

- 引数を持たない．(引数の型は void 型とする．)
- 戻り値を持たない．(戻り値の型は void 型とする．)
- 積分対象関数 $f(x)$ を説明するメッセージを標準出力に出力する副作用を持つ．例えば，

$$f(x) = \frac{4}{1+x^2}$$

の場合は，

```
void description()
{
  printf("f(x)=4.0 / (1.0 + x * x )");
  return;
}
```

のように定義されていればよい．

- description 内では，標準入力は行なってはならない．(scanf 文による値の読み込みは，行なってはならない．)

以上の要求仕様は最低限のものである．もちろん，できるだけ高精度な解を求めるようなプログラムや，できるだけ高速に解を求めるようなプログラムであることが望ましい．また，積分対象となる関数の種類や，積分区間などによっては積分値が存在しない場合があるかもしれない．そのような判定も行えればより良いプログラムである．さらに，定積分計算アルゴリズムによっては，精度を制御するためのパラメータの値の取り得る範囲に制限がある可能性もある．そのような場合にも適切にメッセージを出力するように工夫すると良い．

なお,

$$\int_0^1 \frac{4}{1+x^2} dx = \pi$$

であるので，作成中のプログラムの動作確認に利用するとよい.

TS2-2:定積分 (レポート)
課題 TS2-1 のプログラムの原理，使用法を説明する文書を作成せよ．レポートは,「セメスタ課題報告書の書き方」を参考にすると良い．レポートは A4 のレポート用紙を用いて作成し，所定の提出箱に提出せよ．