システム名：成績管理プログラム
作成者：瀬川　侑
作成日：2018/4/19
最終更新日：2018/5/6
バージョン：1.0
概要：成績を入力しそれらから様々なデータを計算したり、入力したデータを保存したりするプログラム
動作環境：C言語のコンパイラができる環境。LINUX、Visual Studio Community 2017での動作は確認済み
※ただしVisual Studioでは以下の一行をプログラムの一行目に書き足す必要がある
#pragma warning(disable : 4996)
実行方法：
LINUX：端末にてcファイルがあるディレクトリまで移動後、
gcc cファイル名
./a.out
Visual Studio Community 2017：cファイルを開いた後、デバッグなしで開始。初期設定ならCtrl+F5。
必要なインクルードファイル：
stdio.h
string.h
stdlib.h
入力：標準入力、ファイル入力
出力：標準出力、ファイル出力
機能：
（入力）
1. 標準入力での氏名、教科名、点数の入力
2. ファイル指定による標準入力と同様の機能
3. 入力されたデータの修正
4. データの追加
5. 名前、教科名は49文字。人数は50人。教科数は50個まで
6. 点数は100点満点の整数のみ。履修していない場合は負の数を入力（ファイル指定の場合は点数部分が「-」）。
7. 入力ではどの項目でもカンマの使用は禁止

（出力）
1. 入力されたデータから平均点を算出後、標準出力
2. 入力されたデータをもとに全員の欠点数の標準出力
3. 入力されたデータをもとに留年者を標準出力（留年基準は高校課程、大学課程から選択可能）
4. 入力されたデータをもとに点数順で標準出力（並び替えは入力された各教科に加え平均点を基準にすることも可能）
5. ファイル指定による全員の各教科の点数を出力

6. 平均点は少数第一位まで出力
7. 欠点の個数に失点は含まない

（例外処理）

1. 点数標準入力後 100 点を超えているものがあれば訂正を要求する
2. ファイル入力でも同様に訂正を要求する。
3. 入力文字列長が超過した場合は、再入力を促す。

備考：

- 標準入力時の異なった型での入力は考慮していない。
- ファイル入力での点数の欠損、書式外の記述などは考慮していない。
- 出力されたファイルは再度入力ファイルとして使えるように平均点などは出力していない
- 出力されたファイルをcsvファイルとすること、又はカンマ区切りのファイルとして読み込ませることによって表計算ソフトなどで開くと表として開くことができる。
- 入力で指定するファイルは以下の書式にすることによって自分で作ることが可能。

  例）

  name,国語,数学,英語
  高専 太郎,90,80,80
  高専 花子,-,60,65
  高専 次郎,46,60,-

  ※一行目の「name」は適当な文字列（出力ファイルには「name」を使用している）最終行にEOFのみを置くこと。
- limitの値を変更することで人数や科目数の上限を変更することができる
- ファイル名の指定では拡張子（.txt）も書くこと。
- 入出力でのファイル名の指定は必ずプログラムと同じディレクトリに置いておくこと。出力ファイルの場合そのファイルがなければ新しく作成する。

各関数の仕様について

関数名：main
概要（外部仕様）：プログラムの流れを司る関数。関数からの返り値などから分岐を判断する。
引数：なし
返り値：なし
使用する変数：
　グローバル変数
　　なし
　ローカル変数
　　int choices;
内部仕様：switch-case 文を用いて入力メニューと出力メニューを操作し、各関数を実行する。
Caller：なし
Callee：setting, inputpoints, inputfiles, chekcfiles, outputall, outputfaults, outputrep, outputrank, outputfiles, outputmenu

関数名：reset
概要（外部仕様）：すべての変数、配列などの中身に0、又はヌル文字を代入する。プログラム開始時に自動で行うため操作には一切関係しない。
引数：なし
返り値：なし
使用する変数：
グローバル変数

```
#define limit 50

int contents, members, i, j, cont;
unsigned int k;

struct points {
  char content[limit][limit];
  int point[limit];
  int sum;
  float ave;
  int fault;
  int out;
};

struct    mates {
  char member[limit][limit];
  struct points grade[limit];
}data;
```

ローカル変数
なし
内部仕様：forループを複数用いて構造体内の変数をすべて0、0.0、ヌル文字を代入する。forループ終了後に条件式に用いた変数など構造体外の変数に0を代入する。
Caller：main
Callee：なし

関数名：inputmenu
概要（外部仕様）：入力に関する選択肢を数字で選択する。
引数：なし
返り値：int r;
使用する変数：
　グローバル変数
　　なし
　ローカル変数
　　int r;
内部仕様：選択肢を表示し、入力された数値をmainへと返す。
Caller：main
Callee：なし

関数名：setting
概要（外部仕様）：mainで手動入力を選択した場合、名前や教科名を入力するプログラム。
引数：なし
返り値：なし
使用する変数：
グローバル変数

```
#define limit 50
int contents, members, i, j, k;
struct points {
  char content[limit][limit];
  int point[limit];
  int sum;
  float ave;
  int fault;
  int out;
};

struct    mates {
  char member[limit][limit];
  struct points grade[limit];
}data;
```

ローカル変数

```
char contname[limit];
```

内部仕様：人数、科目数を入力後、その回数分 forループを用いて名前、科目名を入力させる。科目名のみ全員分登録する必要があるため、入力された文字列を全員の対応する箇所に代入している。
Caller：main
Callee：なし

関数名：inputfiles
概要（外部仕様）：mainでファイル入力を選択した場合、指定したファイルを読み込み使
　　える状態にするプログラム。
引数：なし
返り値：なし
使用する変数：
　グローバル変数

```
#define limit 50
int contents, members, i, j, cont;
unsigned int k;

struct points {
  char content[limit][limit];
  int point[limit];
  int sum;
  float ave;
  int fault;
  int out;
};

struct    mates {
  char member[limit][limit];
  struct points grade[limit];
}data;
```

　ローカル変数

```
FILE *fi;
char c, fname[limit], p[limit];
```

内部仕様：入力に使用するファイルを標準入力後、そのファイルを開く。一行目を読み
　　飛ばし、二行目以降の一つ目のコンマまでを読み、その文字列を登録。同時
　　にこの登録した人数を数え、登録した人数を覚えておく。科目も同様に一行
　　目を順に読む。ここまで数えた人数×科目数分のforループで二行目以降の点
　　数をコンマ区切りで読み込み、登録する。点数が「-」の場合-1を登録する。
Caller：main
Callee：なし

関数名：chekcinputs
概要（外部仕様）：入力された数値が正しいかを判定する。異常があれば再入力を要求する。
引数：なし
返り値：なし
使用する変数：
グローバル変数

```
#define limit 50
int contents, members, i, j;

struct points {
  char content[limit][limit];
  int point[limit];
  int sum;
  float ave;
  int fault;
  int out;
};

struct    mates {
  char member[limit][limit];
  struct points grade[limit];
}data;
```

ローカル変数
なし
内部仕様：入力によって登録された点数が100点を超えていないかすべての点数を確認し、超えていれば再入力を促す。
Caller：main
Callee：なし

関数名：inputpoints
概要（外部仕様）：手動入力の場合、全員の全科目の点数の点数入力を促す。
引数：なし
返り値：なし
使用する変数：
　グローバル変数

```
#define limit 50
int contents, members, i, j, cont;

struct points {
  char content[limit][limit];
  int point[limit];
  int sum;
  float ave;
  int fault;
  int out;
};

struct    mates {
  char member[limit][limit];
  struct points grade[limit];
}data;
```

　ローカル変数
　　なし
内部仕様：人数分×科目数分のforループですべての点数を入力させる。
Caller：main
Callee：なし

関数名：adds
概要（外部仕様）：addとはadditionの略。学生か科目のどちらを追加するか選択する。
引数：なし
返り値：なし
使用する変数：
　グローバル変数
　　なし
　ローカル変数
　　int r;
内部仕様：switch-case 文を用いてどの項目を追加したいか選択する。
Caller：main
Callee：addmember,addcontent

関数名：addmember
概要（外部仕様）：追加したい学生の名前を入力し、その学生の成績を入力する。
引数：なし
返り値：なし
使用する変数：
　グローバル変数
```
int contents, members, i, j;
unsigned int k;

struct points {
  char content[limit][limit];
  int point[limit];
  int sum;
  float ave;
  int fault;
  int out;
};

struct    mates {
  char member[limit][limit];
  struct points grade[limit];
}data;
```
　ローカル変数
```
int r;
```
内部仕様：追加したい学生の人数を確認後名前を入力。inputpointと同様に追加された学生の分だけforループで入力させる。科目数分×追加した人数分のforループで成績を入力させる。
Caller：adds
Callee：なし

関数名：addcontent
概要（外部仕様）：追加したい科目を入力し、その科目について全員の成績を入力する。
引数：なし
返り値：なし
使用する変数：
　グローバル変数

```
int contents, members, i, j;
unsigned int k;

struct points {
  char content[limit][limit];
  int point[limit];
  int sum;
  float ave;
  int fault;
  int out;
};

struct    mates {
  char member[limit][limit];
  struct points grade[limit];
}data;
```

　ローカル変数

```
int r;
char contname[limit];
```

内部仕様：追加したい科目数を確認後科目名を入力。追加した科目×人数分のforループ
　　ですべての学生の成績入力をさせる。
Caller：adds
Callee：なし

関数名：mods
概要（外部仕様）：modとはmodifyの略。学生名か科目名か点数のどれを修正するか選択する。
引数：なし
返り値：なし
使用する変数：
　グローバル変数
　　なし
　ローカル変数
　　int r;
内部仕様：switch-case 文を用いてどの項目を修正したいか選択する。
Caller：main
Callee：modmember,modcontent,modpoint

関数名：modmember
概要（外部仕様）：修正したい学生の名前を選択し、再入力する。
引数：なし
返り値：なし
使用する変数：
　グローバル変数
```
int members, i;

struct    mates {
  char member[limit][limit];
  struct points grade[limit];
}data;
```
　ローカル変数
```
int r;
```
内部仕様：学生の一覧を表示し、学生の番号で選択。その学生の名前を入力させ、対応した番号の配列に上書きする。
Caller：mods
Callee：なし

関数名：modcontent
概要（外部仕様）：修正したい科目名を選択し、再入力する。
引数：なし
返り値：なし
使用する変数：
　グローバル変数

```
int contents, members, i, j;
unsigned int k;

struct points {
  char content[limit][limit];
  int point[limit];
  int sum;
  float ave;
  int fault;
  int out;
};

struct    mates {
  char member[limit][limit];
  struct points grade[limit];
}data;
```

　ローカル変数

```
int r;
char contname[limit];
```

内部仕様：科目の一覧を表示し、科目の番号で選択。その科目の名前を入力させ、対応した番号の配列に上書きする。
Caller：mods
Callee：なし

関数名：modpoint
概要（外部仕様）：修正したい点数を選択し、再入力する。
引数：なし
返り値：なし
使用する変数：
　グローバル変数
```
int contents, members, i;

struct points {
  char content[limit][limit];
  int point[limit];
  int sum;
  float ave;
  int fault;
  int out;
};

struct    mates {
  char member[limit][limit];
  struct points grade[limit];
}data;
```
　ローカル変数
```
int r, s;
```
内部仕様：学生の一覧を表示し、学生の番号で選択。同様に科目も選択。その点数を入力させ、対応した配列に上書きする。
Caller：mods
Callee：なし

関数名：dels
概要（外部仕様）：delとはdeleteの略。学生か科目のどちらを削除するか選択する。
引数：なし
返り値：なし
使用する変数：
　グローバル変数
　　なし
　ローカル変数
　　int r;
内部仕様：switch-case 文を用いてどの項目を削除したいか選択する。
Caller：main
Callee：delmember,delcontent

関数名：delmember
概要（外部仕様）：削除したい学生を選択する。
引数：なし
返り値：なし
使用する変数：
　グローバル変数

```
int contents, members, i, j;
unsigned int k;

struct points {
  char content[limit][limit];
  int point[limit];
  int sum;
  float ave;
  int fault;
  int out;
};

struct    mates {
  char member[limit][limit];
  struct points grade[limit];
}data;
```

　ローカル変数

```
int r,s;
```

内部仕様：学生の一覧を表示し、学生の番号で選択。その番号以降の学生と点数を詰めるように順番に上書きする。最後に人数を格納している変数から1減らすことによってコピー前の最後の学生の情報が残っていても読み込まれないようにする。
Caller：dels
Callee：なし

関数名：delcontent
概要（外部仕様）:
引数：なし
返り値：なし
使用する変数：
　グローバル変数

```
int contents, members, i, j;
unsigned int k;

struct points {
  char content[limit][limit];
  int point[limit];
  int sum;
  float ave;
  int fault;
  int out;
};

struct    mates {
  char member[limit][limit];
  struct points grade[limit];
}data;
```

　ローカル変数

```
int r,s;
```

内部仕様：科目の一覧を表示し、科目の番号で選択。その番号以降の科目と点数を詰めるように順番に上書きする。最後に科目数を格納している変数から1減らすことによってコピー前の最後の科目の情報が残っていても読み込まれないようにする。
Caller：dels
Callee：なし

関数名：culc
概要（外部仕様）：入力された内容から平均点や欠点数などを自動で計算する。
引数：なし
返り値：なし
使用する変数：
　グローバル変数

```
int contents, members, i, j, cont;

struct points {
  char content[50][50];
  int point[50];
  int sum;
  float ave;
  int fault;
  int out;
};

struct    mates {
  char member[50][50];
  struct points grade[50];
}data;
```

　ローカル変数
　　なし
内部仕様：学生ごとに点数を確認し、正の数なら合計に加算し、負の数なら科目数から除外し合計には加算しない。同時に欠点、失点かどうかも確認し該当するならそれぞれfault(欠点),out(失点)にて数える。一人のカウントが終わると科目数を初期値にリセットし、次の人をカウントする。
Caller：main
Callee：なし

関数名：outputmenu
概要（外部仕様）：出力に関する選択肢を数字で選択する。
引数：なし
返り値：int r;
使用する変数：
　グローバル変数
　　なし
　ローカル変数
　　int r;
内部仕様：選択肢を表示し、入力された数値をmainへと返す。
Caller：main
Callee：なし

関数名：outputall
概要（外部仕様）：入力された全員の全点数と平均点を出力する。
引数：なし
返り値：なし
使用する変数：
　グローバル変数
```
#define limit 50
int contents, members, i, j;

struct points {
  char content[limit][limit];
  int point[limit];
  int sum;
  float ave;
  int fault;
  int out;
};

struct    mates {
  char member[limit][limit];
  struct points grade[limit];
}data;
```
　ローカル変数
　　なし
内部仕様：名前を出力後、教科名と点数を出力。点数が負の数の場合は「履修していません」と出力。各科目の点数forループを用いて出力後、付け加えて平均点を出力。次の学生の出力を開始する。これを人数分のforループで出力。
Caller：main
Callee：なし

関数名：outputfaults
概要（外部仕様）：欠点を持っている学生の名前と欠点個数を表示する。
引数：なし
返り値：なし
使用する変数：
　グローバル変数
```
#define limit 50
int members, i;

struct points {
  char content[limit][limit];
  int point[limit];
  int sum;
  float ave;
  int fault;
  int out;
};

struct    mates {
  char member[limit][limit];
  struct points grade[limit];
}data;
```
　ローカル変数
　　なし
内部仕様：事前に計算しておいた欠点個数を出力する。欠点個数が0の場合は出力しない。
Caller：main
Callee：なし

関数名：outputrep
概要（外部仕様）：repとは留年者(repeater)の略。留年の条件を高校課程、大学課程から選択するとその結果から留年者の名前を表示する。
引数：なし
返り値：なし
使用する変数：
グローバル変数

```
#define limit 50
int members, i;

struct points {
  char content[limit][limit];
  int point[limit];
  int sum;
  float ave;
  int fault;
  int out;
};

struct    mates {
  char member[limit][limit];
  struct points grade[limit];
}data;
```

ローカル変数

```
int r, flag;
```

内部仕様：留年の条件を選択後、全員の欠点個数、平均点、失点の有無から留年者の名前を出力する。もし一人も条件にかからなければ設定したflagがfalseのままのため「留年者はいません」と出力する。
Caller：main
Callee：なし

関数名：outputrank
概要（外部仕様）：科目、または平均点からソートする基準を選択し、その基準に基づいてソート後、点数の降順で表示する。
引数：なし
返り値：なし
使用する変数：
グローバル変数

```
#define limit 50
int contents, members, i, j;

struct points {
  char content[limit][limit];
  int point[limit];
  int sum;
  float ave;
  int fault;
  int out;
};

struct    mates {
  char member[limit][limit];
  struct points grade[limit];
}data;
```

ローカル変数

```
int r, keep, number[limit], rank[limit];
float keepf, rank[limit];
```

内部仕様：科目一覧と項目として平均点を表示し、数字で選択した項目の点数を別の配列(rank)に番号順に移動。同時に学生の番号をnumberという配列に移動。rank内の点数を基準としてソート。rank内の点数を移動時に同じ個所のnumberも移動することにより点数と学生の番号が常に対応するようにする。出力時にソートした学生の番号をもとに名前を呼び出す。同率の場合、一つ前の配列と点数が同じならば同じ順位を出力する処理をする。ただし一番最初の一位のみ比べる相手がいないため通常の処理と同じだが独立して処理を作っている。最初の選択で平均点を選択した場合点数がすべてfloat型なのでその場合、ソート時の一時退避の変数と最初の点数を移す配列をfloat型にしている。
Caller：main
Callee：なし

関数名：outputfiles
概要（外部仕様）：出力するファイルの名前を入力すると、ファイル入力に使うファイルと同じフォーマットでテキストファイルが作成される。
引数：なし
返り値：なし
使用する変数：
グローバル変数

```
#define limit 50
int contents, members, i, j;

struct points {
  char content[limit][limit];
  int point[limit];
  int sum;
  float ave;
  int fault;
  int out;
};

struct    mates {
  char member[limit][limit];
  struct points grade[limit];
}data;
```

ローカル変数

```
FILE *fo;
char fname[limit];
```

内部仕様：表計算ソフトで使用できるようにするためAにあたるところには名前が並ぶ。A1には適当な文字列をいれるためこのプログラムでは「name」としている。その後コンマ区切りですべての科目を出力。二行目以降に名前、各科目の点数を出力する。点数が負の場合は表計算ソフトに入力した際に紛らわしくなくすため数字ではなく「-」を出力している。最後に入力ファイルと同様のフォーマットにするため改行しEOFを移動させる。
Caller：main
Callee：なし