

2021 年度 C++プログラミング II レポート課題 1

S202148

柳澤 快

- 課題内容

- 3月場所の東の力士のそれぞれ番付、名前、成績、対戦相手が記載されているファイルを読み込み、成績のいい順ソートした結果を別のファイルに出力するプログラムを完成させる。

- プログラムの説明

- 主に Record, Table クラスに力士ファイルの読み込みを行う import 関数, 力士の名前、成績、対戦相手を出力する print 関数, 成績の良い順にソートする sort 関数, 結果を result.txt に出力する result 関数が main()関数内で呼ばれているためそれらを用意する。
- import 関数ではファイルの一行ずつの読み込み方とセッターを用いた record への格納方法について問われる
- メンバ関数 kuro(), shiro(), yasumi()ではそれぞれ黒星または不戦敗の数、白星または不戦勝の数、休みの数をそれぞれ数えるためデータメンバ score から1単語ずつ判定していくことが問われる
- split 関数ではイテレータへの理解が求められ” “と”/t”で区切り文字とし文字列を分割する処理が求められる。ネストを作らないように&&を用いてひとつの条件分で完結するよう工夫。

- プログラム

```
● #include <iostream>
● #include <fstream>
● #include <vector>
● #include <iomanip>
●
● class Record
● { //番付, 力士の名前, 成績, 対戦相手を格納するクラス
●     std::string rank, name;
●     std::vector<std::string> score, aite;
●
● public:
●     /* コンストラクタを定義*/
●     Record() = default; //自動生成を依頼
●
●     //成績をカウント
●     int shiro()
●     {
●         int shiro{0};
```

```

•     /* 白星または不戦勝の数を計算*/
•     for (auto e : score)
•     {
•         if (e == "Shiroboshi" || e == "Fusenshou")
•             shiro++;
•     }
•     return shiro;
• }
• int kuro()
• {
•     int kuro{0};
•     /*黒星または不戦敗の数を計算*/
•     for (auto e : score)
•     {
•         if (e == "Kuroboshi" || e == "Fusenpai")
•             kuro++;
•     }
•     return kuro;
• }
• int yasumi()
• {
•     int yasumi{0};
•     /*やすみの数を計算*/
•     for (auto e : score)
•     {
•         if (e == "Yasumi")
•             yasumi++;
•     }
•     return yasumi;
• }
•
• //データメンバの値を取り出すメンバ関数
• std::string get_r()
• { /*rank に対するゲッター*/
•     return rank;
• }

```

```

•   std::string get_n()
•   { /*name に対するゲッター*/
•       return name;
•   }
•   std::vector<std::string> get_s()
•   { /*score に対するゲッター*/
•       return score;
•   }
•   std::vector<std::string> get_a()
•   { /*aite に対するゲッター*/
•       return aite;
•   }
•
•   //データメンバの値を設定するメンバ関数
•   void set_r(std::string r)
•   { /*rank に対するセッター*/
•       rank = r;
•   }
•   void set_n(std::string n)
•   { /*name に対するセッター*/
•       name = n;
•   }
•   void set_s(std::vector<std::string> s)
•   { /*score に対するセッター*/
•       score = s;
•   }
•   void set_a(std::vector<std::string> a)
•   { /*aite に対するセッター*/
•       aite = a;
•   }
•
•   //力士の名前、番付、成績、対戦相手を出力
•   void print()
•   {
•
•       std::cout << "Banzuke:" << rank << std::endl;
•       std::cout << "Shikona:" << name << std::endl;

```

```

•     std::cout << "Seiseki: ";
•     for (auto e : score)
•         std::cout << e << " ";
•     std::cout << std::endl;
•     std::cout << "Taisen_aite: ";
•     for (auto e : aite)
•         std::cout << e << " ";
•     std::cout << std::endl;
• }
• };
•
• class Table
• {
•     //Recordをvectorに格納するクラス
•
•     friend Record; //privateデータメンバへのアクセス許可
•     std::vector<Record> v;
•
• public:
•     void add(Record r)
•     {
•         /*vにrを追加*/
•         v.push_back(r);
•     }
•     void print()
•     {
•         for (auto e : v)
•             e.print();
•     }
•     void sort()
•     {
•         size_t n = v.size();
•         for (size_t i = 0; i < v.size() - 1; i++)
•         {
•             size_t max{i};
•             for (size_t j = i + 1; j < v.size(); j++)

```

```

•      /*勝敗(白星, 不戦勝, 黒星, 不戦敗)によってvをソートする(選択ソート)*/
•      {
•          if (v[max].shiro() > v[j].shiro()) //前後のvの要素を比較し、前の要素より後
の要素の勝ち数が多ければ swap で前後を交換
•              max = j;
•          else if (v[max].shiro() == v[j].shiro() && v[max].kuro() <
v[j].kuro()) //前後の勝ち数が同じ場合、負け数が少なければ swap で前後を交換
•              max = j;
•          std::swap(v[j], v[max]);
•      }
•  }
•  }

•
•
•  void result(std::ofstream &fout)
•  {
•      //出力ファイルに各力士のデータを出力するメンバ関数
•      for (auto e : v)
•      {
•          if (e.yasumi() > 0)
•              fout << std::setw(22) << e.get_r() << std::setw(20) << e.get_n() <<
std::setw(5) << e.shiro() << "Shou" << std::setw(5) << e.kuro() << "Hai"
<< std::setw(5) << e.yasumi() << "Kyu" << std::endl;
•          else
•              fout << std::setw(22) << e.get_r() << std::setw(20) << e.get_n() <<
std::setw(5) << e.shiro() << "Shou" << std::setw(5) << e.kuro() << "Hai"
<< std::endl;
•      }
•  }
•  };
•
•
•  std::vector<std::string> split(std::string s, char sep = ' ', char sep2 =
'¥t')
•  {
•      //文字列から" "と"¥n"の区切りでvに格納
•      std::vector<std::string> v;
•      auto iter{s.begin()}; //sの最初を表すイテレータ

```

```

•   while (iter != s.end()) //テキストが残っている間繰り返す
•   {
•       /*空白(sep=' ')またはタブ(sep2='\\t')で区切られている s の要素を v に格納.
string(iter_first,iter)で iter_first と iter で指定された文字の一部抽出ができる*/
•       auto last = iter;
•       while (last != s.end() && *last != sep && *last != sep2) //seq かつ seq2
        までイテレータを進める
•       ++last;
•       v.push_back(std::string(iter, last)); //分割文字を v へ格納
•       if (last != s.end())
•       ++last;
•       iter = last; //次のループのために last の示すイテレータを設定
•   }
•   return v;
• }
•
• void import(std::ifstream &fin, Table &table)
• { //カスファイルの取り込み
•     Record record;
•     std::string rank, name, score, aite, blank;
•     int n{0};
•     while (fin)
•     {
•         if (n % 5 == 0) //各力士データの一行目を取得
•         {
•             /*fin から rank を取り出し record に格納*/
•             getline(fin, rank);
•             record.set_r(rank);
•         }
•         else if (n % 5 == 1) //各力士データの二行目を取得
•         {
•             /*fin から name を取り出し record に格納*/
•             getline(fin, name);
•             record.set_n(name);
•         }
•         else if (n % 5 == 2) //各力士データの三行目を取得

```

```

●     {
●         /*fin から score を取り出し record に格納. その際 split 関数を用いる*/
●         getline(fin, score);
●         record.set_s(split(score));
●     }
●     else if (n % 5 == 3) //各力士データの四行目を取得
●     {
●         /*fin から aite を取り出し record に格納. その際 split 関数を用いる*/
●         /*record を table に追加*/
●         getline(fin, aite);
●         record.set_a(split(aite));
●         table.add(record); //Table 型に上記で取得した Record 型のデータを格納
●     }
●     else
●     {
●         getline(fin, blank);
●     }
●     n++;
● }
●
● int main()
● {
●     std::ifstream fin("higashi.txt");
●     std::ofstream fout("result.txt");
●
●     Table table;
●     import(fin, table);
●     table.print();
●     table.sort();
●     table.result(fout);
●     return 0;
● }

```

● 実行結果

\$./a.out

Banzuke:Yokozuna

Shikona:Hakuhou

Seiseki: Shiroboshi Shiroboshi Fusenpai Yasumi Yasumi Yasumi Yasumi Yasumi Yasumi
Yasumi Yasumi Yasumi Yasumi Yasumi Yasumi

Taisen_aite: Daieishou Takarafuji Ounoshou

Banzuke:OOzeki

Shikona:Shoudai

Seiseki: Kuroboshi Shiroboshi Shiroboshi Kuroboshi Kuroboshi Kuroboshi Shiroboshi
Shiroboshi Kuroboshi Shiroboshi Shiroboshi Shiroboshi Kuroboshi Kuroboshi Kuroboshi

Taisen_aite: Mitakeumi Ounoshou Takarafuji Hokutofuji Wakatakakage Meisei Shimanoumi
Kiribayama Daieishou Myougiryu Takayasu Takanoshou Terunofuji Takakeishou Asanoyama

Banzuke:OOzeki

Shikona:Takakeishou

Seiseki: Shiroboshi Shiroboshi Kuroboshi Shiroboshi Kuroboshi Shiroboshi Kuroboshi
Shiroboshi Shiroboshi Kuroboshi Shiroboshi Shiroboshi Shiroboshi Shiroboshi Kuroboshi

Taisen_aite: Ounoshou Daieishou Wakatakakage Takarafuji Hokutofuji Shimanoumi Meisei
Myougiryu Mitakeumi Takayasu Kiribayama Okinoumi Asanoyama Shoudai Terunofuji

Banzuke:Sekiwake

Shikona:Terunofuji

Seiseki: Shiroboshi Shiroboshi Shiroboshi Shiroboshi Kuroboshi Shiroboshi Shiroboshi
Kuroboshi Shiroboshi Kuroboshi Shiroboshi Shiroboshi Shiroboshi Shiroboshi Shiroboshi

Taisen_aite: Hokutofuji Wakatakakage Daieishou Meisei Ounoshou Kiribayama Mitakeumi
Takayasu Myougiryu Shimanoumi Takanoshou Tamawashi Shoudai Asanoyama Takakeishou

Banzuke:Komusubi

Shikona:Takayasu

Seiseki: Kuroboshi Shiroboshi Shiroboshi Shiroboshi Shiroboshi Shiroboshi Shiroboshi
Shiroboshi Shiroboshi Shiroboshi Kuroboshi Shiroboshi Kuroboshi Kuroboshi Kuroboshi

Taisen_aite:Meisei Asanoyama Shimanoumi Daieishou Kiribayama Mitakeumi Takarafuji
Terunofuji Ounoshou Takakeishou Shoudai Hokutofuji Wakatakakage Tobizaru Aoiyama

Banzuke:Maegashira_Hittou

Shikona:Takarafuji

Seiseki: Kuroboshi Kuroboshi Kuroboshi Kuroboshi Kuroboshi Shiroboshi Kuroboshi Kuroboshi
Kuroboshi Kuroboshi Shiroboshi Kuroboshi Kuroboshi Shiroboshi Kuroboshi

Taisen_aite: Asanoyama Hakuhou Shoudai Takakeishou Takanoshou Daieishou Takayasu
Mitakeumi Hokutofuji Wakatakakage Shimanoumi Tochinoshin Ounoshou Okinoumi
Kiribayama

Banzuke:Maegashira_Nimaime

Shikona:Hokutofuji

Seiseki: Kuroboshi Shiroboshi Kuroboshi Shiroboshi Shiroboshi Shiroboshi Shiroboshi

Kuroboshi Shiroboshi Kuroboshi Shiroboshi Kuroboshi Shiroboshi Shiroboshi Kuroboshi

Taisen_aite: Terunofuji Takanoshou Asanoyama Shoudai Takakeishou Myougiryu Ounoshou

Shimanoumi Takarafuji Daieishou Mitakeumi Takayasu Ichinojou Ryuden Wakatakakage

Banzuke:Maegashira_Sanmaime

Shikona:Meisei

Seiseki: Shiroboshi Kuroboshi Shiroboshi Kuroboshi Kuroboshi Shiroboshi Shiroboshi

Shiroboshi Kuroboshi Fusenshou Shiroboshi Shiroboshi Kuroboshi Shiroboshi Shiroboshi

Taisen_aite: Takayasu Mitakeumi Kiribayama Terunofuji Asanoyama Shoudai Takakeishou

Takanoshou Wakatakakage Endo Ichinojou Tobizaru Daieishou Tamawashi Tsurugishou

Banzuke:Maegashira_Yonmaime

Shikona:Kiribayama

Seiseki: Kuroboshi Shiroboshi Kuroboshi Shiroboshi Kuroboshi Kuroboshi Shiroboshi

Kuroboshi Kuroboshi Shiroboshi Kuroboshi Shiroboshi Kuroboshi Shiroboshi Shiroboshi

Taisen_aite: Shimanoumi Endo Meisei Okinoumi Takayasu Terunofuji Asanoyama Shoudai

Takanoshou Tamawashi Takakeishou Ounoshou Tochinoshin Daieishou Takarafuji

Banzuke:Maegashira_Gomaime

Shikona:Endo

Seiseki: Kuroboshi Kuroboshi Kuroboshi Shiroboshi Shiroboshi Kuroboshi Shiroboshi

Shiroboshi Shiroboshi Fusenpai Yasumi Yasumi Yasumi Yasumi Yasumi

Taisen_aite: Myougiryu Kiribayama Tamawashi Ichinojou Shimanoumi Okinoumi Tochinoshin

Kagayaki Kotonowaka Meisei

Banzuke:Maegashira_Rokumaime

Shikona:Tamawashi

Seiseki: Kuroboshi Shiroboshi Shiroboshi Kuroboshi Shiroboshi Kuroboshi Shiroboshi

Shiroboshi Kuroboshi Kuroboshi Kuroboshi Kuroboshi Kuroboshi Kuroboshi Kuroboshi

Taisen_aite: Okinoumi Tochinoshin Endo Myougiryu Kagayaki Ichinojou Kotonowaka

Houshouryu Chiyonokuni Kiribayama Wakatakakage Terunofuji Mitakeumi Meisei

Terutsuyoshi

Banzuke:Maegashira_Nanamaime

Shikona:Tochinoshin

Seiseki: Kuroboshi Kuroboshi Shiroboshi Shiroboshi Shiroboshi Kuroboshi Kuroboshi

Kuroboshi Kuroboshi Kuroboshi Shiroboshi Shiroboshi Shiroboshi Shiroboshi Kuroboshi

Taisen_aite: Ichinojou Tamawashi Kotonowaka Kagayaki Okinoumi Tobizaru Endo Midorifuji

Ryuden Akiseyama Chiyotairyu Takarafuji Kiribayama Hidenoumi Takanoshou

Banzuke:Maegashira_Hachimaime

Shikona:Kotonowaka

Seiseki: Kuroboshi Shiroboshi Kuroboshi Kuroboshi Kuroboshi Shiroboshi Kuroboshi

Shiroboshi Kuroboshi Kuroboshi Kuroboshi Shiroboshi Shiroboshi Fusenshou Kuroboshi

Taisen_aite: Kagayaki Chiyonokuni Tochinoshin Tobizaru Houshouryu Midorifuji Tamawashi

Okinoumi Endo Ichinojou Ryuden Chiyoshouma Terutsuyoshi Toyoyama Daiamami

Banzuke:Maegashira_Kyumaime

Shikona:Chiyonokuni

Seiseki: Shiroboshi Kuroboshi Shiroboshi Shiroboshi Shiroboshi Shiroboshi Shiroboshi

Kuroboshi Shiroboshi Kuroboshi Kuroboshi Shiroboshi Fusenpai Yasumi Yasumi

Taisen_aite: Tobizaru Kotonowaka Midorifuji Houshouryu Ryuden Kagayaki Akiseyama

Ichinojou Tamawashi Okinoumi Daiamami Myougiryu Tsurugishou

Banzuke:Maegashira_Jumaime

Shikona:Midorifuji

Seiseki: Shiroboshi Shiroboshi Kuroboshi Kuroboshi Kuroboshi Kuroboshi Kuroboshi

Shiroboshi Shiroboshi Kuroboshi Kuroboshi Kuroboshi Kuroboshi Kuroboshi Shiroboshi

Taisen_aite: Houshouryu Chiyotairyu Chiyonokuni Aoiyama Tobizaru Kotonowaka Chiyoshouma

Tochinoshin Daiamami Hidenoumi Kagayaki Ichinojou Kaisei Tsurugishou Okinoumi

Banzuke:Maegashira_Juichimaime

Shikona:Chiyotairyu

Seiseki: Shiroboshi Kuroboshi Shiroboshi Kuroboshi Kuroboshi Kuroboshi Shiroboshi

Shiroboshi Kuroboshi Shiroboshi Kuroboshi Kuroboshi Kuroboshi Kuroboshi Shiroboshi

Taisen_aite: Ryuden Midorifuji Akiseyama Terutsuyoshi Aoiyama Houshouryu Tsurugishou

Kotoekou Toyoyama Kagayaki Tochinoshin Hidenoumi Daiamami Kaisei Shimanoumi

Banzuke:Maegashira_Junimaime

Shikona:Akiseyama

Seiseki: Shiroboshi Shiroboshi Kuroboshi Shiroboshi Kuroboshi Shiroboshi Kuroboshi

Kuroboshi Kuroboshi Shiroboshi Kuroboshi Shiroboshi Shiroboshi Kuroboshi Kuroboshi

Taisen_aite: Kotoshouhou Terutsuyoshi Chiyotairyu Ryuden Chiyoshouma Kaisei Chiyonokuni

Daiamami Aoiyama Tochinoshin Tsurugishou Houshouryu Toyoyama Kotoekou Daieishou

Banzuke:Maegashira_Jusanmaime

Shikona:Terutsuyoshi

Seiseki: Kuroboshi Kuroboshi Shiroboshi Shiroboshi Shiroboshi Kuroboshi Shiroboshi

Shiroboshi Shiroboshi Kuroboshi Kuroboshi Kuroboshi Kuroboshi Shiroboshi Shiroboshi

Taisen_aite: Aoiyama Akiseyama Kotoekou Chiyotairyu Toyoyama Tsurugishou Hidenoumi

Tobizaru Chiyoshouma Daiamami Kaisei Kotoshouhou Kotonowaka Kagayaki Tamawashi

Banzuke:Maegashira_Juyonmaime

Shikona:Kotoekou

Seiseki: Shiroboshi Shiroboshi Kuroboshi Shioboshi Kuroboshi Shioboshi Shioboshi

Kuroboshi Kuroboshi Kuroboshi Kuroboshi Shioboshi Shioboshi Shioboshi Kuroboshi

Taisen_aite: Chiyoshouma Toyoyama Terutsuyoshi Daiamami Hidenoumi Tokushoryu Ryuden

Chiyotairyu Tsurugishou Aoiyama Tobizaru Kaisei Chiyohomaru Akiseyama Kagayaki

Banzuke:Maegashira_Jugomaime

Shikona:Toyoyama

Seiseki: Shioboshi Kuroboshi Kuroboshi Kuroboshi Kuroboshi Kuroboshi Shioboshi

Shioboshi Shioboshi Kuroboshi Kuroboshi Kuroboshi Kuroboshi Fusenpai Yasumi

Taisen_aite: Tsurugishou Kotoekou Daiamami Daishoumaru Terutsuyoshi Hidenoumi Kaisei

Chiyoshouma Chiyotairyu Ishiura Aoiyama Ryuden Akiseyama Kotonowaka

Banzuke:Maegashira_Jurokumaime

Shikona:Kaisei

Seiseki: Kuroboshi Shioboshi Shioboshi Shioboshi Shioboshi Kuroboshi Kuroboshi

Kuroboshi Shioboshi Kuroboshi Shioboshi Kuroboshi Shioboshi Shioboshi Kuroboshi

Taisen_aite: Hidenoumi Akua Tsurugishou Chiyoshouma Daiamami Akiseyama Toyoyama

Chiyonoou Kagayaki Houshouryu Terutsuyoshi Kotoekou Midorifuji Chiyotairyu Tobizaru

result.txt

Sekiwake	Terunofuji	12Shou	3Hai	
Maegashira_Sanmaime	Meisei	10Shou	5Hai	
Komusubi	Takayasu	10Shou	5Hai	
OOzeki	Takakeishou	10Shou	5Hai	
Maegashira_Nimaime	Hokutofuji	9Shou	6Hai	
Maegashira_Kyumaime	Chiyonokuni	8Shou	5Hai	2Kyu
Maegashira_Juyonmaime	Kotoekou	8Shou	7Hai	
Maegashira_Jurokumaime	Kaisei	8Shou	7Hai	
Maegashira_Jusanmaime	Terutsuyoshi	8Shou	7Hai	
Maegashira_Yonmaime	Kiribayama	7Shou	8Hai	
Maegashira_Nanamaime	Tochinoshin	7Shou	8Hai	
Maegashira_Junimaime	Akiseyama	7Shou	8Hai	
OOzeki	Shoudai	7Shou	8Hai	
Maegashira_Hachimaime	Kotonowaka	6Shou	9Hai	
Maegashira_Juichimaime	Chiyotairyu	6Shou	9Hai	
Maegashira_Gomaime	Endo	5Shou	5Hai	5Kyu

Maegashira_Rokumaimae	Tamawashi	5Shou	10Hai	
Maegashira_Jumaimae	Midorifuji	5Shou	10Hai	
Maegashira_Jugomaimae	Toyoyama	4Shou	10Hai	1Kyu
Maegashira_Hittou	Takarafuji	3Shou	12Hai	
Yokozuna	Hakuhou	2Shou	1Hai	12Kyu

- 考察

- Nishi.txt 内の引退した力士を一番うしろへソートする機能を追加

```
#include <iostream>
#include <fstream>
#include <vector>
#include <iomanip>

class Record
{ //番付, 力士の名前, 成績, 対戦相手を格納するクラス
    std::string rank, name;
    std::vector<std::string> score, aite;

public:
    /* コンストラクタを定義*/
    Record() = default; //自動生成を依頼

    //成績をカウント
    int shiro()
    {
        int shiro{0};
        /* 白星または不戦勝の数を計算*/
        for (auto e : score)
        {
            if (e == "Shiroboshi" || e == "Fusenshou")
                shiro++;
        }
        return shiro;
    }
    int kuro()
    {
```

```

    int kuro{0};
    /*黒星または不戦敗の数を計算*/
    for (auto e : score)
    {
        if (e == "Kuroboshi" || e == "Fusenpai")
            kuro++;
    }
    return kuro;
}

int yasumi()
{
    int yasumi{0};
    /*やすみの数を計算*/
    for (auto e : score)
    {
        if (e == "Yasumi")
            yasumi++;
    }
    return yasumi;
}

int intai()
{
    int intai{0};
    for (auto e : score)
    {
        if (e == "Intai")
            intai++;
    }
    return intai;
}

//データメンバの値を取り出すメンバ関数
std::string get_r()
{ /*rank に対するゲッター*/
    return rank;
}

```

```

std::string get_n()
{ /*name に対するゲッター*/
    return name;
}

std::vector<std::string> get_s()
{ /*score に対するゲッター*/
    return score;
}

std::vector<std::string> get_a()
{ /*aite に対するゲッター*/
    return aite;
}

//データメンバの値を設定するメンバ関数
void set_r(std::string r)
{ /*rank に対するセッター*/
    rank = r;
}

void set_n(std::string n)
{ /*name に対するセッター*/
    name = n;
}

void set_s(std::vector<std::string> s)
{ /*score に対するセッター*/
    score = s;
}

void set_a(std::vector<std::string> a)
{ /*aite に対するセッター*/
    aite = a;
}

//力士の名前、番付、成績、対戦相手を出力
void print()
{
    std::cout << "Banzuke:" << rank << std::endl;
    std::cout << "Shikona:" << name << std::endl;
}

```

```

        std::cout << "Seiseki:";
        for (auto e : score)
            std::cout << e << " ";
        std::cout << std::endl;
        std::cout << "Taisen_aite:";
        for (auto e : aite)
            std::cout << e << " ";
        std::cout << std::endl;
    }
};

class Table
{
    //Recordをvectorに格納するクラス

    friend Record; //private データメンバへのアクセス許可
    std::vector<Record> v;

public:
    void add(Record r)
    {
        /*vにrを追加*/
        v.push_back(r);
    }
    void print()
    {
        for (auto e : v)
            e.print();
    }
    void sort()
    {
        size_t n = v.size();
        for (size_t i = 0; i < v.size() - 1; i++)
        {
            size_t max{i};
            for (size_t j = i + 1; j < v.size(); j++)

```



```

        /*勝敗(白星, 不戦勝, 黒星, 不戦敗)によってvをソートする(選択ソート)*/
        {
            if (v[max].shiro() > v[j].shiro()) //前後のvの要素を比較し、前の要素より
            後の要素の勝ち数が多ければ swap で前後を交換

                max = j;
            else if (v[max].shiro() == v[j].shiro() && v[max].kuro() >
            v[j].kuro()) //前後の勝ち数が同じ場合、負け数が少なければ swap で前後を交換

                max = j;
            std::swap(v[j], v[max]);
        }
        //引退者を一番後ろへソート
        if (v[max].intai() > 0)
        {
            v.push_back(v[max]);
            v.erase(v.begin() + max);
        }
    }
}

void result(std::ofstream &fout)
{
    //出力ファイルに各力士のデータを出力するメンバ関数
    for (auto e : v)
    {
        if (e.intai() > 0)
            fout << "Intai Player===== (" << std::setw(22)
            << e.get_r() << std::setw(20) << e.get_n() << std::setw(5) << e.shiro() <<
            "Shou" << std::setw(5) << e.kuro() << "Hai" << std::setw(5) << e.yasumi() <<
            "Kyu"

                << ")" << std::endl;
            else if (e.yasumi() > 0)
                fout << std::setw(22) << e.get_r() << std::setw(20) << e.get_n()
            << std::setw(5) << e.shiro() << "Shou" << std::setw(5) << e.kuro() << "Hai" <<
            std::setw(5) << e.yasumi() << "Kyu" << std::endl;
            else

```

```

        fout << std::setw(22) << e.get_r() << std::setw(20) << e.get_n()
<< std::setw(5) << e.shiro() << "Shou" << std::setw(5) << e.kuro() << "Hai" <<
std::endl;
    }
}
};

std::vector<std::string> split(std::string s, char sep = ' ', char sep2 =
'¥t')
{
    //文字列から" "と"¥n"の区切りでvに格納
    std::vector<std::string> v;
    auto iter{s.begin()}; //sの最初を表すイテレータ
    while (iter != s.end()) //テキストが残っている間繰り返す
    {
        /*空白(sep=' ')またはタブ(sep2='¥t')で区切られているsの要素をvに格納.
string(iter_first,iter)でiter_firstとiterで指定された文字の一部抽出ができる*/
        auto last = iter;
        while (last != s.end() && *last != sep && *last != sep2) //sepかつsep2
までイテレータを進める
            ++last;
        v.push_back(std::string(iter, last)); //分割文字をvへ格納
        if (last != s.end())
            ++last;
        iter = last; //次のループのためにlastの示すイテレータを設定
    }
    return v;
}

void import(std::ifstream &fin, Table &table)
{ //カスファイルの取り込み
    Record record;
    std::string rank, name, score, aite, blank;
    int n{0};
    while (fin)
    {

```

```

    if (n % 5 == 0) //各力士データの一行目を取得
    {
        /*fin から rank を取り出し record に格納*/
        getline(fin, rank);
        record.set_r(rank);
    }
    else if (n % 5 == 1) //各力士データの二行目を取得
    {
        /*fin から name を取り出し record に格納*/
        getline(fin, name);
        record.set_n(name);
    }
    else if (n % 5 == 2) //各力士データの三行目を取得
    {
        /*fin から score を取り出し record に格納. その際 split 関数を用いる*/
        getline(fin, score);
        record.set_s(split(score));
    }
    else if (n % 5 == 3) //各力士データの四行目を取得
    {
        /*fin から aite を取り出し record に格納. その際 split 関数を用いる*/
        /*record を table に追加*/
        getline(fin, aite);
        record.set_a(split(aite));
        table.add(record); //Table 型に上記で取得した Record 型のデータを格納
    }
    else
    {
        getline(fin, blank);
    }
    n++;
}

int main()
{

```

```
std::ifstream fin("nishi.txt");
std::ofstream fout("result.txt");

Table table;
import(fin, table);
table.print();
table.sort();
table.result(fout);
return 0;
}
```

● 実行結果

OOzeki	Asanoyama	10Shou	5Hai	
Maegashira_Nimaime	Wakatakakage	10Shou	5Hai	
Komusubi	Mitakeumi	8Shou	7Hai	
Sekiwake	Takanoshou	8Shou	7Hai	
Komusubi	Daieishou	8Shou	7Hai	
Maegashira_Hachimaime	Tobizaru	10Shou	5Hai	
Maegashira_Kyumaime	Houshouryu	8Shou	7Hai	
Maegashira_Junimaime	Aoiyama	11Shou	4Hai	
Maegashira_Jugomaime	Hidenoumi	10Shou	5Hai	
Maegashira_Jurokumaime	Daiamami	9Shou	6Hai	
Maegashira_Juyonmaime	Tsurugishou	9Shou	6Hai	
Maegashira_Jusanmaime	Chiyoshouma	8Shou	7Hai	
Maegashira_Rokumaime	Ichinojou	7Shou	8Hai	
Maegashira_Yonmaime	Myougiryu	7Shou	8Hai	
Maegashira_Jumaime	Ryuden	6Shou	9Hai	
Maegashira_Nanamaime	Kagayaki	6Shou	9Hai	
Maegashira_Sanmaime	Shimanoumi	4Shou	11Hai	
Maegashira_Hittou	Ounoshou	4Shou	11Hai	
Maegashira_Gomaime	Okinoumi	3Shou	12Hai	
Maegashira_Juichimaime	Kotoshouhou	1Shou	6Hai	8Kyu
Intai	Player=====	(Yokozuna
Kakuryu	0Shou 0Hai 10Kyu)			

- 感想

- 今回のレポート課題の中で最も苦戦したのは split 関数を実装することである。主にイテレータの理解が必要不可欠であるが、なかなか期待通りの実行結果が得られず試行錯誤した。