**TUGAS KECIL 1**

**IF2211 – Strategi Algoritma**

Penyelesaian Permainan Kartu 24

dengan Algoritma Brute Force



Disusun oleh

Frankie Huang 13521092

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**T.A. 2022/2023**

1. **Algoritma Brute Force**

Algoritma *brute force* adalah pendekatan paling *straightforward* untuk memecahkan suatu persoalan. Pada *source file*, algoritma brute force digunakan untuk mencari semua jawaban yang mungkin dan untuk melakukan pencarian untuk mengubah integer menjadi string dan sebaliknya.

Misalkan terdapat 4 buah bilangan: *a*, *b*, *c*, dan *d*; dan 4 buah operasi yang dapat dilakukan: “+”, “-”, “\*”, dan “/”. Untuk operasi 4 buah bilangan, terdapat 5 jenis urutan pengoperasian yang dapat dilakukan, yaitu:

1. ((a *op1* b) *op2* c) *op3* d
2. (a *op1* (b *op2* c)) *op3* d
3. (a *op1* b) *op2* (c *op3* d)
4. a *op1* ((b *op2* c) *op3* d)
5. a *op1* (b *op2* (c *op3* d))

Namun, urutan pengoperasian 1 dan 5 serupa karena keempat bilangan akan dipermutasikan, sehingga salah satu urutan pengoperasian dapat diabaikan. Kemudian, lakukan nested loop 3 kali, masing-masing untuk tiap *op*, sehingga untuk keempat bilangan yang berbeda, terdapat maksimal buah urutan pengoperasian yang akan diperiksa oleh program.

Untuk algoritma *brute force* yang digunakan untuk melakukan pencarian, cara yang digunakan cukup mudah dimengerti. Pencarian dilakukan secara linear hingga ditemukan dan jika tidak ditemukan, maka ada kesalahan pada input pengguna yang akan mengakibatkan pada berhenti berjalannya program.

1. **Implementasi pada Bahasa Pemrograman C++**

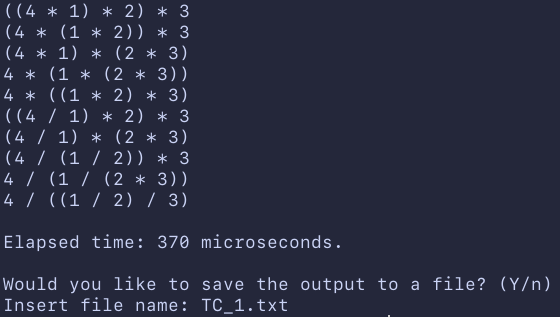
Pada *source file*, terdapat 4 buah variabel global dan 3 buah struktur data global yang dijelaskan pada tabel di bawah ini

| Nama | Fungsi |
| --- | --- |
| char input[16] | Menyimpan input pengguna dalam bentuk *array of character* |
| int integer[4] | Menyimpan keempat bilangan dalam bentuk *integer* |
| int total | Menyimpan total jawaban yang mungkin didapatkan |
| vector<float> numbers | Menyimpan keempat bilangan dalam bentuk *float* |
| vector<string> answers | Menyimpan jawaban yang didapatkan |
| set<vector<float>> permutations | Mengecek apakah permutasi sudah pernah ditemukan sebelumnya |

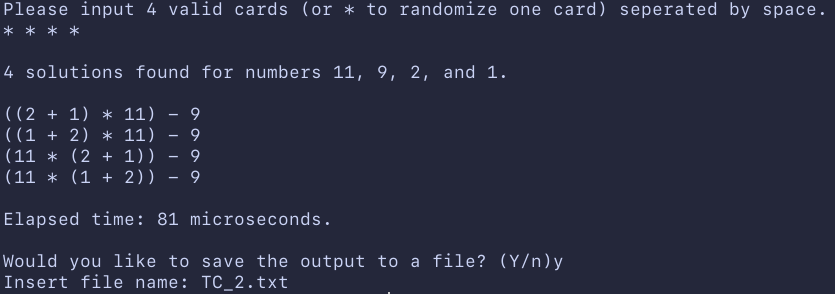
Selain itu, terdapat juga fungsi-fungsi yang terdapat pada *source file*, yaitu

| Nama | Fungsi |
| --- | --- |
| float convert(char c) | Mengubah *input* dalam bentuk *char* menjadi *float* serta mengubah karakter “\*” menjadi angka random |
| void readInput() | Menerima input pengguna dan melakukan pengecekan input pengguna |
| float operate(float a, float b, int op) | Melakukan operasi antara *float* a dan *float* b sesuai dengan nilai *op* |
| string insertOperator(int op) | Mengembalikan string sesuai dengan nilai *op* |
| void findResult(int i, int j, int k, int type) | Menyusun urutan pengoperasian yang mungkin dan menyimpannya pada answers |
| void swap(int a, int b) | Melakukan perubahan nilai pada indeks *a* dan *b* pada numbers |
| void findNumbers(int n) | Melakukan algoritma *backtracking* untuk mencari seluruh permutasi bilangan |
| int main() | Memanggil fungsi lain serta menghitung waktu yang dibutuhkan dan penyimpanan jawaban pada file |

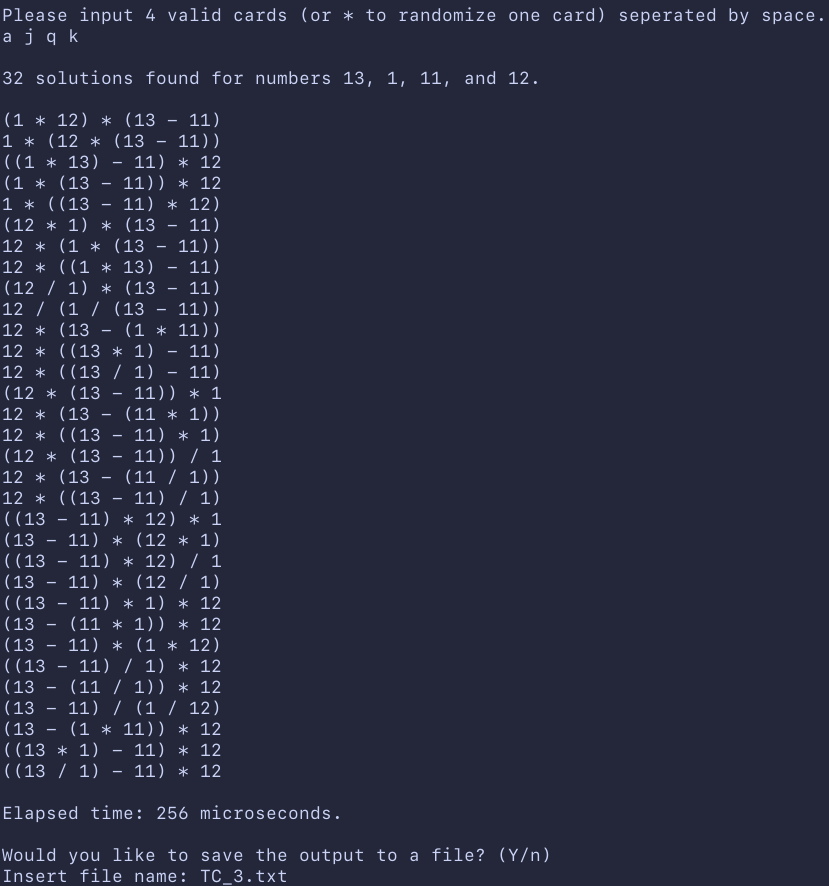
1. **Eksperimen**
2. Kasus 4 buah bilangan (1 2 3 4)



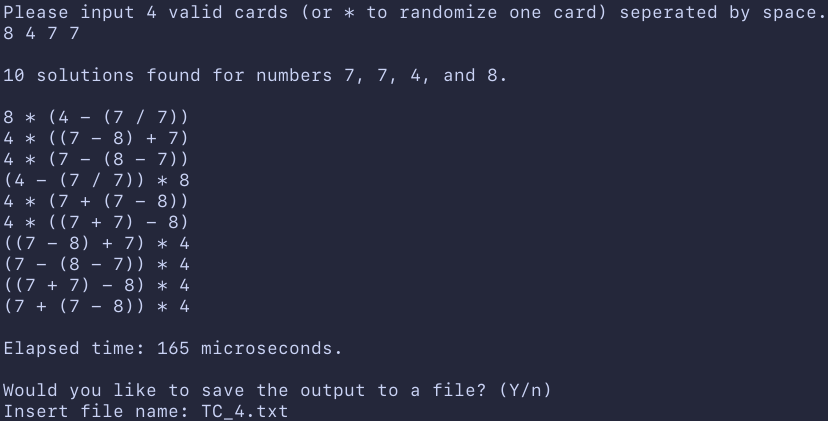
1. Kasus 4 buah bilangan acak (\* \* \* \*)



1. Kasus karakter A, J, Q, dan K (a j q k)



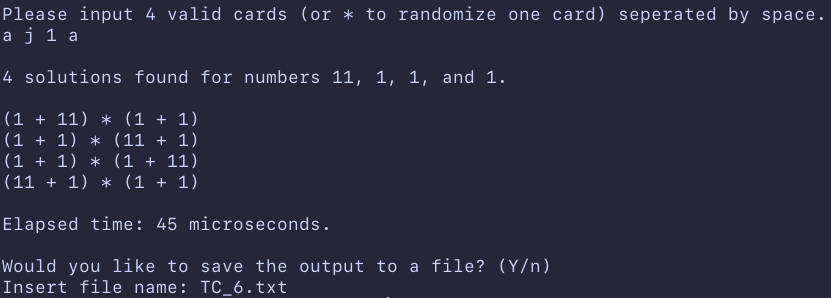
1. Kasus 2 bilangan serupa (8 4 7 7)



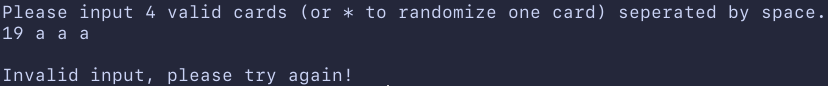
1. Kasus 2 bilangan acak (4 7 \* \*)



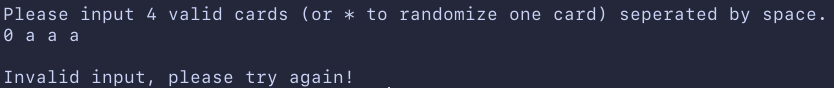
1. Kasus karakter A dan nilai 1 (a j 1 a)



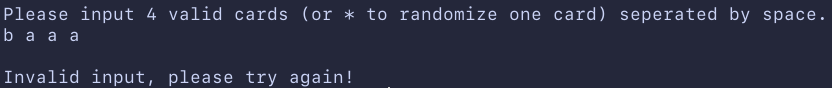
1. Kasus angka tidak valid (19 a a a)



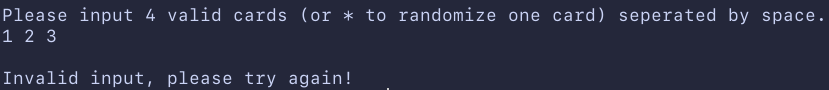
1. Kasus angka tidak valid (0 a a a)



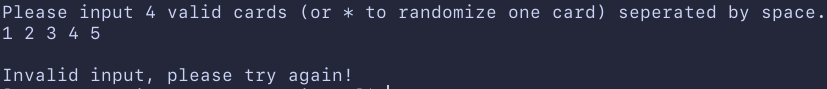
1. Kasus karakter tidak valid (b a a a)



1. Kasus input kurang dari 4 (1 2 3)



1. Kasus input lebih dari 5 (1 2 3 4 5)



1. **Pranala**

Pranala spesifikasi tugas : <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/Tucil1-Stima-2023.pdf>

Pranala repository : <https://github.com/frankiehuangg/Tucil1_13521092>