|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Practicum Final Exam – Odd Semester Year 2021/2022** | | | | | | |
| **Subject** | | | **COMP6047 – Algorithm and Programming** | | |  |
| **Class** | **:** | **09CB** | | **Start Date** | **: 03 December 2021** |
| **Lecturer** | **:** | **D6439 - Yesun Utomo S.Kom., M.T.I** | | **Start Time** | **: 08.00** |
| **End Date** | **: 03 December 2021** |
| **End Time** | **: 10.00** |

**PERATURAN UJIAN:**

*Exam Regulations:*

* Mahasiswa tidak diperbolehkan berdiskusi dan/atau bekerja sama dengan peserta ujian lainnya

*Student is not allowed to discuss and/or work together with other exam participants*

* Mahasiswa tidak diperbolehkan untuk membuka dan menyalin dari **BUKU** atau **CATATAN**, **VIDEO** dari pengajar (recording kelas, VBL, Youtube, dsb) dan **REFERENSI** lainnya

*Student isn't allowed to open and copy from any resources such as notes, videos (class recording, VBL, Youtube, etc) and other references*

* Mahasiswa tidak diperbolehkan membuka dan menyalin jawaban dari internet (google, stackoverflow, dsb)

*Student isn't allowed to open and copy answer from the internet (google, stackoverflow, etc)*

* Asisten **BERHAK** memberi nilai 0 **(NOL)** bagi peserta ujian yang melakukan segala bentuk kecurangan

*Assistant is able to give 0 (ZERO) score for exam participant who does any cheating actions*

* Kumpulkan jawaban tepat pada waktunya, apabila terlambat mengumpulkan maka jawaban tidak akan dikoreksi dan nilai mahasiswa adalah 0

*Submit the answer on time, if not, then the answer will not be checked, and the students will receive 0 (ZERO)*

* Bila Anda tidak membaca peraturan ini, maka Anda dianggap telah membaca dan menyetujuinya

*If you have missed to read these regulations, so you are considered to have read and agreed on it*



**SOFTWARE YANG DIGUNAKAN:**

*Software will be used:*

* Dev-C++ 5.11

**FILE YANG DIKUMPULKAN:**

*File must be collected:*

* CPP

**PERHATIAN!**

*Attention!*

* Bagi yang mengerjakan tidak sesuai dengan soal, maka akan diberikan nilai **NOL (0)**

*For those who do not work in accordance with the exam case will be marked as* ***ZERO (0)***

* Bagi yang mengerjakan tidak sesuai dengan software dan versi yang telah ditetapkan, maka akan tetap dikoreksi dengan software dan versi yang telah ditetapkan

*For those who do not work in accordance with the software and specific version will be corrected by the predefined software and version*

* Kompres semua jawaban yang akan diunggah. Pastikan format pengumpulan nama file dan ekstensi sesuai dengan format berikut: **[NIM]-[NAMA].zip**

*Compress all file that will be uploaded. Make sure the format for collecting file name and extension according to the following format:* ***[NIM]-[NAME].zip***

**Important Notes:**

1. Do not code only on the main function, **code every module on its own separate function**
2. You are only **allowed** using **Merge Sort/ Quick Sort Algorithm**
3. You are **allowed** using a **struct** to store all your variable that needed

**Soal**

*Case*

**Stiim**

**Stiim** is a famous Game Store that sells various of interesting and popular games. To improve the business process, the manager hired you as a programmer to build a program to help the Store to manage the games and the orders from the customer. Your task is to create the program using **C Programming Language** based on the description below:

* In the first time, the program will **read the file** which the **name** is ‘***games.txt’*** using the following format:

|  |
| --- |
| **Code#Name#Type#Price#Quantity** |

* If the file does not exist, **do not insert any data**.
* If the file exists, but there are no content inside it, **do not** **insert any data**.
* If the file exists and there are contents inside the file, **insert the data** and stored it in **an array**.
* The program consists of **4 menus**:

1. Add Game
2. View Game
3. Purchase Game
4. Exit

A picture containing shape

Description automatically generated **Figure 1. Main Menu**

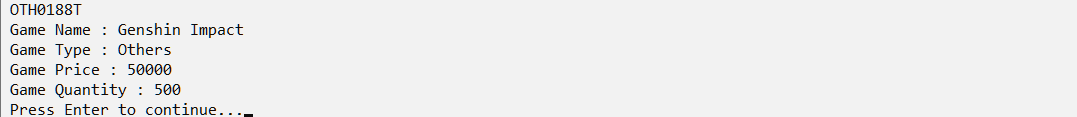
* The program will ask the user to input the **menu number**, which must be **between 1** and **4**
* If the user chooses **menu** **1** (‘**Add Game’**), then:
* The program will ask the user to input **Game details**, which consists of:
* **Game Type**, which must be **between 1** and **3**
* **Game Name**, which **length** must **between 3** and **20 characters (whitespace included)**
* **Game Price**, which must be **between 10000** and **10000000.**
* **Game Quantity**, which must be **between 1** and **200000.**
* If the inputs are valid, **generate Game Code** using following formula:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Format: **AAAYYYYB**  **A** 🡪 **First 3 *Capitalized* Characters of the game type**  **Y** 🡪 **random number between 0** and **9**  **B 🡪 Last *Capitalized* Characters of the game name**  Example:  Game Name: Genshin Impact  Game Type: Others   |  |  | | --- | --- | | AAA | **OTH** | | Random Number 1 | **8** | | Random Number 2 | **5** | | Random Number 3 | **3** | | Random Number 4 | **7** | | B | **T** |   Game Code: **OTH8537T** |

* **Add the data to an array** and show the details of the inserted game containing(**Game Code, Game Type, Game Price, and Game Quantity**):
* Then display a message ‘**Press enter to continue…**’

A picture containing chart

Description automatically generated**Figure 2. Add Game Details Menu**

**Figure 3. Preview Game Details Menu**

* If the user chooses **menu** **2** (‘**View Game’**), then:
* The program will ask the user to input **Menu** that wants to view, which must be **between 0** and **4**

A picture containing shape

Description automatically generated**Figure 4. View Game Menu**

* If the user chooses a menu **between** **1** and **3**, then show the game list **according to the selected game type**
* If the user chooses **menu 4**, then show the game list from **all game type**.
* If the user chooses **0**, then **redirect** **user** to the **previous menu.**
* If the **selected** list is **empty**, then show the message ‘**No Game Registered’**

A picture containing shape

Description automatically generated**Figure 5. View Game Menu (List is empty)**

* Otherwise, show the **sorted** data based on **game code** with **ascending** order

Table

Description automatically generated with low confidence**Figure 6. View Game Menu (List has items)**

* If the user chooses **menu** **3** (‘**Purchase Game**’), then:
* If the list is **empty**, then show the message ‘**No game Registered in Stiim’**



**Figure 7. Purchase Game Menu (List is empty)**

* Otherwise:
* The program will ask the user to input **Game** that wants to purchase, which must be **between 1** and **4**

A picture containing shape

Description automatically generated**Figure 8. Purchase Game Menu (Wrong menu number)**

* If the user chooses a menu **between** **1** and **3**, then show the game list **according to the selected game type**
* If the user chooses **menu 4**, then show the game list from **all game type**
* If the **selected** list is **empty**, then:
* Show the message ‘**No Game of this type is registered’**

A picture containing shape

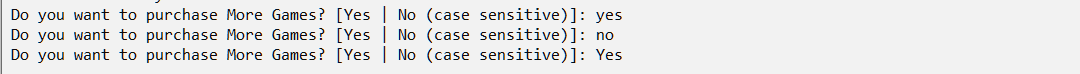
Description automatically generated**Figure 9. Purchase Game Menu (Selected List is empty)**

* Otherwise:
* Show the **sorted** data based on **game code** with **ascending** order
* Ask the user to input the **purchase index**, which must be **between 1** and **count of the selected list**
* Ask the user to input the **quantity**, which must be **between 1** and **the selected game’s quantity**

Table

Description automatically generated with medium confidence**Figure 10. Purchase Game Menu (Selected List has items)**

* After that, **add the data to a cart,** and display a message (’**Game successfully added to Cart!**’)
* If the **quantity of the purchased game is 0,** then **remove the purchased game from the list.**
* The program will ask the user to input **to** **purchase another game**, which must be either ‘**Yes**’ or ‘**No**’ (**case sensitive**)

**Figure 11. Purchase Game Menu (Purchase More After Buying Games)**

* If the user chooses ‘**Yes**’, then the program will **ask again to input Game that wants to purchase**
* Otherwise:
* The program will show the **bill payment**
* The **bill payment** will show the **details of payment** which consist of **Total Game Purchased, Game Name**, **Quantity**, **Game Price**, **Subtotal**, **Total Price**, **Tax** (**10%**), and **Grand Total Price**
* The **Subtotal**, **Total Item**, **Total Price**, **Tax** (**10%**), and **Grand** **Total Price** obtained using the following formula:

|  |
| --- |
| **Subtotal = Quantity \* Game Price** |

|  |
| --- |
| **Game [1]**  **Game [2]**  **Game […]**  **Game [n]**  **--------------------------------------------- +**  **Total Games** |

|  |
| --- |
| **Subtotal [1]**  **Subtotal [2]**  **Subtotal […]**  **Subtotal [n]**  **--------------------------------------------- +**  **Total Price** |

|  |
| --- |
| **Tax (10%) = Total Price \* 10 / 100** |

|  |
| --- |
| **Grand Total Price = Total Price + Tax (10%)** |

A picture containing shape

Description automatically generated**Figure 12. Bill Payment Information**

* If the user chooses **menu** **4** (‘**Exit**’), then:
* The program will **save** the menu list to the **file** which the **name** is ‘***games.txt’*** using the following format:

|  |
| --- |
| **Code#Name#Type#Price#Quantity** |

* After that, the program will be closed

**Run the program for more details!**