EF234201 Data Structure (F)

Midterm Exam

Starting date:	17 April 2024
Deadline:	24 April 2024, 23:59 WIB. **Penalty: 0.15% of grade/minute of tardiness.**
Exam type:	Open, Individual Project
Send to:	MM Irfan Subakti <yifana@gmail.com> CC to Hammuda Arsyad <hammudaarsyad@gmail.com>, Frederick Yonatan Susanto <frederickyonatan111@gmail.com>, Rakha Fathin Izzan Consetta <rakhaconsetta@gmail.com> & Muhammad Izzul Sinar Mahadhika <zulsinar031@gmail.com> with the subject: **EF234201_DS(F)_MID_StudentID_Name**
File type and format:	A **zip** file containing all of the **.cpp** (*source files*) & the **declaration**
Filename format:	**EF234201_DS(F)_MID_StudentID_Name.ZIP**

Instruction
Please do these steps as in the following.

1. Complete the program below in C++, namely `01_stack_[your_name].cpp`. [EASY] [**10 points**]
   Implement the following functions.
   a. The `pop()` function, this function has a function to pop the most upper item in the stack.
   b. The `push()` function, this function is to push an item to the stack
   c. The `peek()` function, this function is to peek or see the utmost/the top of the item in the stack, not remove it.
   d. The `isEmpty()` function, this function is for checking whether the stack is empty or not.

Code:

```
#include <bits/stdc++.h>

using namespace std;

#define MAX 1000

class Stack {
    int top;

public:
    int a[MAX]; // Maximum size of Stack

    Stack() {
        top = -1;
    }
    bool push(int x);
    int pop();
    int peek();
    bool isEmpty();
};

int main()
{
    class Stack s;
    s.push(10);
    s.push(20);
    s.push(30);
    cout << s.pop() << " popped from stack\n";
    return 0;
}
```

2. Complete the program below in C++, namely `02_deque_[your_name].cpp`. [MEDIUM] [**30 points**]

Code:

```
#include <iostream>

using namespace std;

typedef struct snode_t { // Node
    char data, direction;
    int step;
    struct snode_t *next;
    struct snode_t *prev;
} node;

typedef struct dlist_t { // Deque
    unsigned _size;
    node *_head;
} linkedList;

bool slist_isEmpty(linkedList *list) {
    return (list->_head == NULL);
}

void slist_pushBack(linkedList *list, char direction, char data,
int step) {
      // Push back new data
}

void transverse (linkedList *myList) {
      // Transverse the list
}

int main() {
    linkedList *myList = new linkedList;
    myList->_head = NULL;

    char direction, data;
    int step;
    do {
        cin>>direction>>step>>data;
        slist_pushBack(myList, direction, data, step);
    } while (direction != '0');

    transverse(myList);

    return 0;
}
```

Input:

Direction Step Data

- `direction` is the direction to transverse next; `value`: 'b' for backward or 'f' for forward
- `step` is the step to the direction; for example, `b 5` means transverse 5 nodes backwards

- `data` is the data for the node
- The program will stop when the `direction` is `0`

Sample input:

```
f 3 s
f 3 k
f 4 r
b 1 t
f 3 d
f 3 t
b 5 u
b 2 a
0 0 0
```

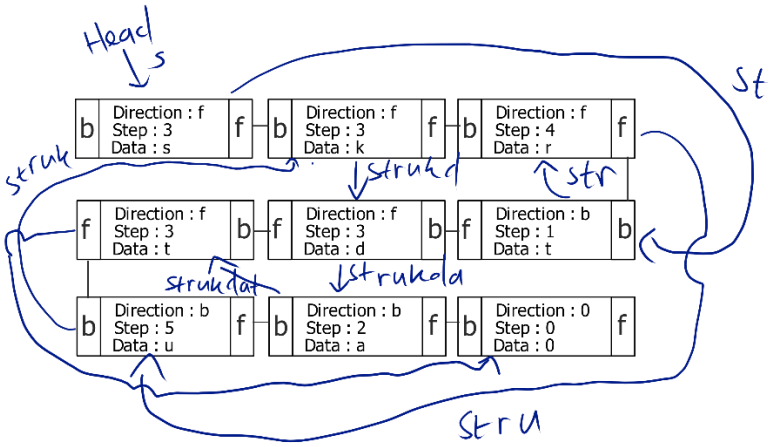
Output:

Print the data which has been taken from the transversal.

Sample output:

```
strukdat
```

Illustration:



3. Make a program namely `03_odds_even_[your_name].cpp`. Sort the number that has been input. Output the even number in ascending order, then the odd one in descending order.
   PS: You must use the linked list on this problem. Don't use arrays. [MEDIUM] [**30 points**]

   Input:

   The first line is *N*, how many test cases. Then the following *N* row is the number that has to be sorted

   Output:

   Every number which already sorted by the criteria above. Output line by line.

Sample input:

```
10
4
32
34
543
3456
654
567
87
6789
98
```

Sample output:

```
4
32
34
98
654
3456
6789
567
543
87
```

4. Code the following task on a program that reverses string using a linked list named `04_reverse_string_[your_name].cpp`. [EASY] [**15 points**]

Sample input:

```
practice makes a man perfect
```

Sample output:

```
perfect man a makes practice
```

5. Complete the following code, namely `05_remove_string_[your_name].cpp` where `remove_duplicate()` is a function to remove the node with the same data on the list. [EASY] [**15 points**]

Code:

```
//linkedlist

void remove_duplicate (linkedList * myList) {
      //remove duplicate data
}

int main() {
    linkedList *myList = new linkedList;
    myList->_head = NULL;
    int data;
    do {

        cin>> data;
        slist_pushBack(myList, data);
    } while (data != -1);
    print_linkedList(myList);
    remove_duplicate(myList);
    print_linkedList(myList);
    return 0;
}
```

Sample input:

```
1
2
3
4
1
5
6
3
2
-1
```

Sample output:

```
1 2 3 4 1 5 6 3 2 -1
1 2 3 4 5 6
```

6. To avoid plagiarism/cheating, every student needs to pledge and declare, then she/he must submit her/his **signed pledge and declaration** as in the following. Failing to do so will result in getting a 0 (zero) grade. Attach the **scanned**/**photo** of your *declaration* in your report.

   "By the name of Allah (God) Almighty, herewith I pledge and truly declare that I have solved the midterm exam by myself, didn't do any cheating by any means, didn't do any plagiarism, and didn't accept anybody's help by any means. I am going to accept all of the consequences by any means if it has proven that I have done any cheating and/or plagiarism."

   [Place, e.g., Surabaya], [date, e.g., 24 April 2024]

   <Signed>

   [Full name, e.g., Kurnia Pujawati]
   [StudentID, e.g., 5025231xxx]

7. **ZIP** the files of `01_stack_[your_name].cpp`, `02_deque_[your_name].cpp`, `03_odds_even_[your_name].cpp`, `04_reverse_string_[your_name].cpp`, `05_remove_string_[your_name].cpp` and your **declaration** (e.g., `Declaration.PDF`) into 1 (one) only .ZIP file, namely **`EF234201_DS(F)_MID_StudentID_Name.ZIP`**. Send this .**ZIP** file to `yifana@gmail.com` and CC the **TA's emails**.

8. Have a wonderful day, guys! Good luck! 😊