# National Institute of Technology Calicut
# Department of Computer Science and Engineering
# Second Semester B. Tech.(CSE)
# CS1092E Program Design Laboratory
# Assignment #2
# Submission deadline (on or before): 12/03/2024, 08:00 PM

**Naming Conventions for Submission**

- Submit a single ZIP (.zip) file (do not submit in any other archived formats like .rar, .tar, .gz). The name of this file must be

  `<FIRST-NAME>_<ROLLNO>.zip`

  (Example: *LAXMAN_BxxyyyyCS.zip*). DO NOT add any other files (like temporary files, input files, etc.) except your source code, into the zip archive.

- The source codes must be named as

  `<FIRST-NAME>_<ROLLNO>_<PROGRAM-NUMBER>.c`

  (For example: *LAXMAN_BxxyyyyCS_1.c*). If you do not conform to the above naming conventions, your submission might not be recognized by our automated tools, and hence will lead to a score of 0 marks for the submission. So, make sure that you follow the naming conventions.

**Standard of Conduct**

- Violation of academic integrity will be severely penalized. Each student is expected to adhere to high standards of ethical conduct, especially those related to cheating and plagiarism. Any submitted work **MUST BE an individual effort**. Any academic dishonesty will result in zero marks in the corresponding exam or evaluation and will be reported to the department council for record keeping and for permission to assign F grade in the course. The department policy on academic integrity can be found at: http://cse.nitc.ac.in/sites/default/files/Academic-Integrity_new.pdf.

**General Instructions**

- Programs should be written in C language.
- Check your programs with sufficiently large values of inputs with in the range as specified in the question.
- Global and/or static variables should not be used in your program.

**Question 1-** Write a 'C' program to input names of 'N' different individuals and store them in an array and to find the largest name among them. Use dynamic memory allocation to allocate space efficiently for each string such that the space allocated for each string in the array will be proportional to the size of the string. Make sure to release the allocated memory using the function free() before ending the program.

**Input format**: The first line contains an integer indicating the number of names N in the array. The next N lines contain N Strings of varying lengths. Assume that each string is of length at most 50.

**Output format:** The largest String of the array.

**Sample Input 1:**
5
Anuj K P
Binitha Nair
Johan Thomas K
Pankaj Singh
Aysha Hayya S

**Sample Output 1:**
Johan Thomas K

**Question 2-** Given the name and total marks of 'N' students and an integer r, find the student with 'r'th rank. Write a C program to get the details of the 'r'th ranked student. Use the function malloc() for allocating space for the 'N' students, and use structures for storing student details. Make sure to release the allocated memory using the function free() before ending the program.

**Input Format**:
The first line read the number of students, N.
The Next N lines reads the Name and total marks of students ($0 \leq$ total_mark $\leq 100$).
The next line contains an integer r.

**Output Format:**
Details of 'r'th ranked student

**Sample Input1:**
4
Aarathy 67
Anagha 89
Manu 96
Yadav 77
1

**Sample Output1:**
Manu 96

**Question 3 -** Write a menu driven C program to create a singly linked list of integers and then sort the linked list in ascending order using O(1) additional space.

Your program must contain the following functions:

- Main() - This function repeatedly reads a character ‘i’, ‘p’,’s’ or ‘e’ from the terminal and calls the sub-functions appropriately, as mentioned in Input format, until character ‘e’ is entered.
- CreateNode(num) - This function creates a new node that stores integer num as data, and a pointer next that points to NULL. This procedure returns the address of the new node.
- InsertNode(Head,num) - This function is used to insert the node with number num at the end of the list. Here, the pointer Head has the address of the first node of the linked list. This function will call CreateNode function to create a new node with number num.
- Print(Head) - This function takes the address of the first node of the linked list as argument and prints all the nodes present in the linked list.
- Sort(Head) - This function sorts the linked list in ascending order using O(1) additional space.

Input/Output Format:

- Each line in the input starts with a character from {i,p,s,e}
- Character ‘i’ followed by an integer num. Inserts a new node with num as data at the end of the linked list using InsertNode() function.
- Character ‘s’ is used to sort the linked list using Sort() function..
- Character ‘p’ is used to print all the data in the linked list using Print() function. Print NULL if the list is empty. The output should be printed on a separate line.
- Character ‘e’ is to exit from the program.

**Sample Input/Output 1:**

```
i 1
i 0
i 2
i 1
i 0
i 2
i 1
s
p
0 0 1 1 1 2 2
i 1
p
0 0 1 1 1 2 2 1
s
p
0 0 1 1 1 1 2 2
e
```

**Question 4 -** In a land of digits and integers, a wizard named Numbo needs assistance organizing his magical numbers using a doubly linked list.

Numbo needs you to build a menu-driven C program that contains the following functions:

- MAIN() - repeatedly reads a character ‘i’, ‘p’,’r’ or ‘e’ from the terminal and calls the sub-functions appropriately, as mentioned in Input format, until character ‘e’ is entered.
- CREATE NODE(num) - creates a new node that stores integer num as data, and a pointer next that points to NULL. This procedure returns the address of the new node.
- LIST_INSERT(head, x) - inserts node x to the end of the linked list whose address of the first node is stored in head.
- INITIAL_LIST(head) - prints the data in list L in order, starting from the head, separated by a space.
- REVERSE_LIST(head)- A function to reverse the list whose address of the first node is stored in head.

**Input/Output Format:**

- Each line in the input starts with a character from {i,p,r,e}
- Character ‘i’ followed by an integer num. Inserts a new node with num as data at the end of the linked list using LIST_INSERT() function.
- Character ‘r’ is used to reverse the linked list using REVERSE_LIST() function..
- Character ‘p’ is used to print all the data in the linked list using INITIAL_LIST() function. Print NULL if the list is empty. The output should be printed on a separate line.
- Character ‘e’ is to exit from the program.

**Sample Input/Output 1**

```
i 3
i 7
i 10
i 15
p
3 7 10 15
r
p
15 10 7 3
i 17
p
15 10 7 3 17
e
```

**Question 5 -** Write a menu-driven C program for circular linked lists. Your program contains a function to print the Next Greater Element for every element present in the linked list. The Next Greater Element for an element 'x' is the first element on the right side of 'x' in the linked list, which is greater than 'x'. For the largest element in the linked list, the Next Greater Element is considered to be -1.

Your program must contain the following functions:

- Main() - repeatedly reads a character ‘i’, ‘p’,’g’ or ‘e’ from the terminal and calls the sub-functions appropriately, as mentioned in Input format, until character ‘e’ is entered.
- CreateNode(num) - creates a new node that stores integer num as data, and a pointer next that points to NULL. This procedure returns the address of the new node.
- InsertNode(Head, num) - This function is used to insert the node with number num at the end of the list (in the order starting from the node that Head is pointing to).This function will call CreateNode function.
- Print(Head) - This function takes the address of the node stored in Head and prints all the nodes present in the linked list, starting from Head.
- NGE(num) - This function takes the address of the node stored in p and prints the next greater element for the node containing the data num.

Assume that there are no two nodes in the linked list with the same data value.

**Input/Output Format:**

- Each line in the input starts with a character from {i,p,g,e}
- Character ‘i’ followed by an integer num. Inserts a new node with num as data at the end of the linked list using InsertNode() function.
- Character ‘g’ followed by an integer num. Prints the next greater element in the list for the node containing the integer num.
- Character ‘p’ is used to print all the elements in the linked list using Print() function. Print NULL if the list is empty. The output should be printed on a separate line.
- Character ‘e’ is to exit from the program.

**Test case1:**

```
i 1
i 2
i 3
i 4
i 5
i 6
g 1
2
g 2
3
g 3
4
g 4
5
g 5
6
g 6
 -1
e
```

**Question 6 :** Design and develop a Restaurant Reservation System where customers can reserve tables upon arrival in a fair and organized manner. Upon entering the restaurant, customers provide their name, mobile number, and arrival time to be

added to the reservation list. The system maintains the order of arrival, ensuring fairness in seating arrangements. It offers functionality to view the next booking request without altering the queue. Tables are assigned to customers based on their position in the reservation list. Additionally, customers have the option to cancel their reservations, allowing others to take their place. This system ensures an equitable process for managing table reservations, promoting customer satisfaction, and also displays the order of table reservations on a screen in the restaurant.

Keep the Input and Output format as shown in the Sample Input and Output.

**Sample Input & Output**

Menu:
1. Add Reservation
2. View Next Booking Request
3. Cancel Reservation
4. Display Reservation Queue
5. Exit

Enter your choice: 1
1. Customer Name: Alice Smith
2. Mobile Number: 987-654-3210
3. Arrival Time: 18:45

**Output:**
Reservation added successfully for Alice Smith.

Menu:
1. Add Reservation
2. View Next Booking Request
3. Cancel Reservation
4. Display Reservation Queue
5. Exit

Enter your choice: 2

**Output:**
Next Booking Request: Alice Smith (18:45)

Menu:
1. Add Reservation
2. View Next Booking Request
3. Cancel Reservation
4. Display Reservation Queue
5. Exit

Enter your choice: 3
Reservation to cancel:
Alice Smith

**Output:**
Reservation for Alice Smith canceled successfully.

Menu:
1. Add Reservation
2. View Next Booking Request
3. Cancel Reservation
4. Display Reservation Queue
5. Exit

Enter your choice: 4

**Output:**
Reservation Queue is empty.

Menu:
1. Add Reservation
2. View Next Booking Request
3. Cancel Reservation
4. Display Reservation Queue
5. Exit

Enter your choice: 5

**Output:**
Exiting…

**Question 7:** Design and develop a menu driven program that utilizes the stack data structure to efficiently manage browser history for a web browser. The stack data structure must be implemented using **linked lists**. The application should allow users to navigate through web pages, go back to previously visited pages, and clear their browsing history. Your task is to outline the functionality and design of this application, specifying how the stack data structure will be employed to achieve these operations effectively.

Your program should be implemented using the following functions:

a. Visit_New_Page()
   - When a user visits a new web page, it is added to the browsing history.
   - This action pushes the URL of the new page onto the stack.

b. Go_Back()
   - Users can go back to the previously visited web page.

- This action pops the top URL from the stack, effectively navigating back in the browsing history.

c. Clear_History()
- Users can clear their browsing history, removing all URLs from the stack.
- This action empties the stack, effectively clearing the browsing history.

Keep the Input and Output format as shown in the Sample Input and Output.

**Sample Input and Output :**

Menu:
1. Visit New Page
2. Go Back
3. Display History
4. Exit
Enter your choice: 1
Enter URL: www23333
**Output:**
Page ': www23333' visited successfully.

Menu:
1. Visit New Page
2. Go Back
3. Clear History
4. Display History
5. Exit
Enter your choice: 1
Enter URL: kkkk32k
**Output**
Page 'kkkk32k' visited successfully.

Menu:
1. Visit New Page
2. Go Back
3. Clear History
4. Display History

5. Exit
Enter your choice: 1
Enter URL: ijkl234
**Output**
Page ' ijkl234' visited successfully.

Menu:
1. Visit New Page
2. Go Back
3. Clear History
4. Display History
5. Exit
Enter your choice: 2
**Output**
Navigating back to page ' kkkk32k '.

Menu:
1. Visit New Page
2. Go Back
3. Clear History
4. Display History
5. Exit
Enter your choice: 3
**Output**
History Cleared.

Menu:
1. Visit New Page
2. Go Back
3. Clear History
4. Display History
5. Exit
Enter your choice: 4
**Output**
Browsing History:

Menu:
1. Visit New Page

2. Go Back
3. Clear History
4. Display History
5. Exit
Enter your choice: 5
**Output**
Exiting..