**1) User Input and Replace String Template “Hello <<UserName>>, How are you?”**

a. **I/P** -> Take User Name as Input. *Ensure UserName has min 3 char*

b. **Logic** -> Replace <<UserName>> with the proper name

c. **O/P** -> Print the String with User Name

package com.bridgelabz.programs;

import com.bridgelabz.util.Utility;

public class Replace {

public static void main(String[] args) {

Stirng str1,str2,str3,str4;

utility u = new.utility();

System.out.print("Enter String");

str1 = u.inputString();

System.out.print("Enter name you want to Replace");

str2 = u.inputString();

System.out.print("Enter name you want to Replace with");

str3 = u.inputString();

str4 = str1.Replace(str2,str3);

System.out.println("New String:" +str4);

}

}

**2. Simulate Stopwatch Program**

a. Desc -> Write a Stopwatch Program for measuring the time that elapses between

the start and end clicks

b. I/P -> Start the Stopwatch and End the Stopwatch

c. Logic -> Measure the elapsed time between start and end

d. O/P -> Print the elapsed time.

Import java.util.Concurrent.Timeunit;

Class Timeutil

{

public static void main(String[] args) throughs Interrupted Exception {

long startTime = System.nanoTime()

TimeUnit.SECOND.sleep(5);

Long endTime = System.nanoTime();

Long timeElapsed = endTime – startTime;

System.out.println(“Execution time in nanoseconds:” +timeElapsed);

System.out.println(“Execution time in millisecond:” + timeElapsed/1000000);

}

}

**Intermediate**

1. **Cross Game or Tic-Tac-Toe Game**

Import java.util.scanner;

Public class Tic Tac Toc

{

Public int counter;

Public char Posn[] = new char[10];

Private char player;

Public static void main (string args[])

{

String ch;

Tic Tac Toc Toe = new Tic Tac Toc();

Do{

Toe.new Board();

Toe.play();

System.out.println(“would you like to play again(“Enter ‘Yes’)?”);

Scanner .in = new Scanner (System.in)

Ch = in.nextline();

System.out.println(“ch value is “+ch);

} while(ch. Equal (“Yes”));

}

Public void new Board()

{

Char Posndef[] = { ‘0’ , ‘1’ , ‘2’ , ‘3’, ‘4’, ‘5’, ‘6’, ‘7’, ‘8’, ‘9’};

Int I;.

Counter = 0;

Player = ‘x’;

For(i=1; i<10;, i++) Posn[i] = Posndef[i] ;

Current Board();

}

Public satring current Board()

{

S O P(“\n \n”);

S O P(“\n \n”);

S O P(“\n\n\t\t” + Posn[1] + “1” + Posn[2] + “1” + Posn[3]);

S O P(“\t\t 1 1 “);

S O P(“\t\t\_\_\_1\_\_\_1\_\_\_);

S O P(“\n\n\t\t” + Posn[4] + “1” + Posn[5] + “1” + Posn[6]);

S O P(“\t\t 1 1 “);

S O P(“\t\t\_\_\_1\_\_\_1\_\_\_);

S O P(“\n\n\t\t” + Posn[7] + “1” + Posn[8] + “1” + Posn[9]);

S O P(“\t\t 1 1 “);

S O P(“\t\t 1 1 “);

S O P(“\n \n”);

Return “current Board”;

}

Public void play()

{

Int spot;

Char blank= ‘ ‘ ;

S O P(“Player” + getplayer() + “will gofirst and be the letter ‘x’);

do{

current Board();

S O P(“\n\n player” + getplayer() + “choose a Posn”);

Boolean PosTaken = true;

While(PosTaken){

Scanner in = new Scanner (System . in);

Spot = in.nextInt();

PosTaken = check Posn(Spot);

If(PosTaken == false)

Posn[Spot] = get player();

}

S O P(“Nice move”);

Current Board();

Next player();

} while (Check winner() == blank);

}

Public char check winner()

{

Char winner= ‘ ‘ ;

If(Posn[1] == ‘x’ \*\* Posn[2] == ‘x’ \*\* Posn[3] == ‘x’) winner = ‘x’;

if(Posn[4] == ‘x’ \*\* Posn[5] == ‘x’ \*\* Posn[6] == ‘x’) winner = ‘x’;

If(Posn[7] == ‘x’ \*\* Posn[8] == ‘x’ \*\* Posn[9] == ‘x’) winner = ‘x’;

If(Posn[1] == ‘x’ \*\* Posn[4] == ‘x’ \*\* Posn[7] == ‘x’) winner = ‘x’;

If(Posn[2] == ‘x’ \*\* Posn[5] == ‘x’ \*\* Posn[8] == ‘x’) winner = ‘x’;

If(Posn[3] == ‘x’ \*\* Posn[6] == ‘x’ \*\* Posn[9] == ‘x’) winner = ‘x’;

If(Posn[1] == ‘x’ \*\* Posn[5] == ‘x’ \*\* Posn[9] == ‘x’) winner = ‘x’;

If(Posn[3] == ‘x’ \*\* Posn[5] == ‘x’ \*\* Posn[7] == ‘x’) winner = ‘x’;

If (winner == ‘x’)

{

S O P(“player1 wins the game “);

Return winner;

}

If(Posn[1] == ‘0’ \*\* Posn[2] == ‘0’ \*\* Posn[3] == ‘0’) winner = ‘0’;

If(Posn[4] == ‘0’ \*\* Posn[5] == ‘0’ \*\* Posn[6] == ‘0’) winner = ‘0’;

If(Posn[7] == ‘0’ \*\* Posn[8] == ‘0’ \*\* Posn[9] == ‘0’) winner = ‘0’;

If(Posn[1] == ‘0’ \*\* Posn[4] == ‘0’ \*\* Posn[7] == ‘0’) winner = ‘0’;

If(Posn[2] == ‘0’ \*\* Posn[5] == ‘0’ \*\* Posn[8] == ‘0’) winner = ‘0’;

If(Posn[3] == ‘0’ \*\* Posn[6] == ‘0’ \*\* Posn[9] == ‘0’) winner = ‘0’;

If(Posn[1] == ‘0’ \*\* Posn[5] == ‘0’ \*\* Posn[9] == ‘0’) winner = ‘0’;

If(Posn[3] == ‘0’ \*\* Posn[5] == ‘0’ \*\* Posn[7] == ‘0’) winner = ‘0’;

If(winner == ‘0’)

{

S O P(“player2 wins the game “);

Return winner;

}

For (int I = 1 ; i<10 ; i++)// check for Tie

{

If(Posn[i] == ‘x’ || posn [i] == ‘0’){

If(I ==9)

{

Char Draw = ‘0’;

S O P(“Game is statement”);

Return Draw;

}

Continue ;

}

Else

Brake;

}

Return winner;

}

Public Boolean check Posn(int spot)

{

If(posn[spot] = = ‘x’ || Posn[spot] == ‘0’)

{

S O P(“That Posn is already taken, please choose another”);

Return true;

}

}

Public void next player()

{

If(player == ‘x’)

Player = ‘0’

Else player= ‘x’;

}

Public string get Title()

{

Return “ Tic Tac Toc”;

}

Public char getplayer()

{

Return player;

}

}

**Take a string from user at runtime. Check that using those characters of string is it**

**possible to make a palindrome string? If yes then print the palindrome string.**

Import java.util.scanner;

Class Palindrome

{

Public static void main (string args[])

{

String a,b;

String buffer sb;;

Scanner sc = new scanner (System.in);

System.out.println(“enter the string to check if it is a palindrome”);

a = sc.nextline();

sb = new string buffer(a);

b = sb. Reverse().to string();

if (a.equal(b))

{

S O P(“entered string is a palindrome”);

}

Else

{

S O P(“entered string is not a palindrome”);

}

}

}

**For a linked list perform the following functions.**

1. Reverse the link in single link list

Class linkedlist {

Static node head;

Static class node{

Int data;

Node next;

Node (int d)

{

Data = d;

Next = null;

}

}

Node reverse(node node)

{

Node prev;

Node current = node;

Node next = null;

While (current ! = null)

Next = current.next;

Current.next = prev;

Prev = current;

Current = next;

}

Node = prev;

Return node;

{

Void print list (node node)

{

While(node != null){

System.out.print(node.data + “ “ );

Node = node.next;

}

}

Public static void main (String [] arg())

{

Linkedlist list= new linked list();

List.head = new node(85);

List.head.next = new node(15);

List.head.next.next = new node(4);

List.head.next.next.next = new node(20);

S O P(“Given linked list”)

List.println(head);

Head = list.reverse (head);

S O P(“ ”);

S O P(“reversed linked list”);

List.printlist(head);

}

}

1. Reverse the link using recursion

Import java.io.buffered writer;

Import java.io.IoException;

Import java.io.OutputStreamWriter;

Import ava.Util.Scanner;

Public class reverselinked list Recurcive;

{

Static class node{

Public int data;

Public node next;

Public node(int node data){

This.data = node data;

This.next = null;

}

}

Static class linked list(

Public node head;

Public linkedlidt(){

Public this.head= null;

}

Public void insert node(int node data){

Node node = new node(node data);

If(this.head ! = null){

Node.next = head;

}

This.head = node;

}

}

Public static void print singly linkedlist(Node node string sep)throws IoException{

While (node! = node){

S O P(String value of (node.data)+sep);

Node = node.next;

}

}

Static node reverse(Node head){

If(head = = null){

Return head;

}

If (head.next = = null){

Return head;

}

Node new head node = reverse(head.next);

Head.next.next = head;

Head.next=null;

Return new head node;

}

Private static final scanner scanner = new scanner(System.in);

Public static void main(String [] args)throws IoException{

linked list llist = new linked list()

Llist. Insert node(20);

Llist ,insertnode(40;

Llist. Insert node(15);

Llist .insertnode(85);

S O P(“Given linkedlist “);

Pri nt singly linked list (llist head, “ “ );

S O P();

S O P(“Reversed linked list “);

Node llist = reverse (llist.head);

Print singly linkedlist (llist !, “ “ );

Scanner.close();

}

}

1. Reverse the link using stack.

#include<stdio.h>

#include<conio.h>

#define max 50

Struct stock{

Int a [max];

Int top;

}s;

Int main(){

Int list[] ={10,20,30,40};

Int I,size;

Clrscr();

Stop = 1

Printf(“the input list in: “);

Size = size of (list)/size of (list[0]);;

For(I = o ; i<size ; i++) {

Printf(“%d”, list[i]);

}

For(i=0 ; i<size ; I ++){

List[i] = pop();

}

Print(“\n the output list is=”);

For(i=0; i<size ; I ++){

Print(“%d”,list[i]);

}

Getch();

Return 0;

}

Push(int value){

If (stop == max-1){

Printf(“\n stack is full”);

}

Else{

s.top++;

s.a[s.top] = value;

}

Return;

}

Int pop()

{

Int value;

If(s.top == -1)

{

Printf(“\n stack is empty”);

Return;

}

Else

{

Value = s.a[s.top];

s.top -- ;

return value;

}

}