

Task 3: Programming Problems by implementing necessary control statements.

a) Raj and Raju are thick friends. They decided to play a game. Raj is an excellent guy with good communication skills. Raj decided to test his friend raju with few tongue twisters. Raj will utter some sentence told by his friend. The task is replace all the words that has been uttered by raj in the sentence with a special character?

Aim: To solve programming problems by implementing necessary control statements algorithm:

Step 1: Start

Step 2: Take integer variable x.

Step 3: Divide the variable x with (x-1 to 2)

Step 4: Print result.

Program

```
#include <stdio.h>
```

```
#include <string.h>
```

```
int main()
```

```
char
```

Output:
enter a sentence: veltch university is in chennas
enter a character to replace e: a
modified sentence: veltch ~~university~~ is in Chennai.

This program does not print anything so we
will test it by hand you will see what happens
when you print without output and then after this
is done with print without output this will show
what happens and what should be the output of the
program. This is the output of the program

Output:
X Aditya says hi to say
(say I-D) Aditya X Aditya all said says

fibonacci
morpion

calculator calculator
(calculator)

{Gamer}

```
sentence[100];
char replace_char;
printf("Enter a sentence:");
fgetws(sentence, 100, stdin);
printf("Enter a character to replace:");
scanf("%c", &replace_char);

for (int i=0; i<strlen(sentence);
     i++) { if (sentence[i] == replace_char)
    {
        sentence[i] = '#';
    }
}
printf("Modified sentence: %s", sentence);
return 0;
}
```

- 6) write a program that generates a random number and asks the user to guess what the number is. If the user's guess is higher than the random number, If the user's guess is lower than the random number, The program should use a loop that repeats until the user correctly guesses?

Algorithm:

1. Generate a random number between a specified range.
2. Ask the user to enter a guess for the random number.
3. Check if the guess is equal to the random number.
If it is, exit the loop and print a message.
4. If the guess is higher than the random number,
print a message saying "Too high, try again".
5. If the guess is lower than the random number, print
a message saying "Too low, try again".
6. Repeat steps 3-6 until the user correctly guesses
the number.
7. End the program.

Program

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main()
{
    srand(time(0));
    int random Number = rand() % 100 + 1;
```

output

Give the number between 1 and 100: 80. Too low, try again.

Guess the number between 1 and 100: go Too high, try again

Guess the number between 1 and 100: 81 Too Low, try again

Guess the number between 1 and 100: 82 Too low, try again
I have too good of a guess.

Guess the number between 1 and 100: 83

Congratulations! You guessed the number 83.

After numerous reboots the fault seems to have disappeared.

Import and export problems

reduced 11

missed ab his

प्रिया

<2.016> shabu

<deadbeats> ~~deadbeats~~

~~Chmidt > Shulman~~

(J) Dian J.

(O) 2017 (F) 1000

11.00011(0) base = 110001 mod 101 find

```

int guess;
do {
    printf("Guess the number between 1 and 100:");
    scanf("%d", &guess);
    if (guess > random number) {
        printf("Too high, try again.\n");
    }
    else if (guess < random Number) {
        printf("Too low, try again.\n");
    }
} while (guess != random Number);
printf("Congratulations! You guessed the number %d.\n",
random Number); return 0;
}

```

Result.- Thus the program is written and executed successfully.

VFL TECH - CSE	
EX NO.	92
PERFORMANCE (5)	8
RESULT AND ANALYSIS (3)	7
VIVA VOCE (3)	7
RECORD (4)	8
TOTAL (15)	24
SIGN WITH DATE	