

Virtual Lab

Ashwera Hasan

Roll No:16010124107

Class/Div: C2-2

Program 1(screenshot)

The screenshot shows the Virtual Lab interface for a Selection Sort simulation. The browser address bar displays `cse02-iiith.vlabs.ac.in/exp/arrays/simulation.html`. A notification box from `cse02-iiith.vlabs.ac.in` states "code running is over!". The interface is divided into three main sections:

- Initialize:** Contains a form to "Enter Array Size" (set to 5) and "Enter Values Manually" (12, 23, 42, 10, 32). Buttons for "Start" and "Next" are visible.
- Step Execution:** Displays the C code for Selection Sort:

```
int main() {
    int i, j, size, key;
    int A[size];
    for (i = 1; i < size; i++)
    {
        key = A[i];
        j = i - 1;
        while (j >= 0 && A[j] > key)
        {
            A[j+1] = A[j];
            j--;
        }
        A[j+1] = key;
    }
    return 0;
}
```
- Code Output:** Shows the array state: 10, 12, 23, 32, 42. It includes radio buttons for "Sorted", "Unsorted", and "Key Position".

The Windows taskbar at the bottom shows the date as 29-03-2025 and time as 20:12.

2

The screenshot shows the Virtual Lab interface for a Matrix Multiplication simulation. The browser address bar displays `cse02-iiith.vlabs.ac.in/exp/arrays/simulation.html`. A notification box from `cse02-iiith.vlabs.ac.in` states "Code Running Is Over!". The interface is divided into three main sections:

- Initialize:** Contains a form to "Enter Matrix Size" (2 x 2). Buttons for "OK", "Generate Values For B", "Start", and "Next" are visible.
- Step Execution:** Displays the C code for matrix multiplication:

```
int main() {
    int i, j, k;
    int matA[i][j];
    int matB[j][k];
    int matMult[i][k];
    int p, q, r;
    for (p = 0; p < i; p++)
    {
        for (r = 0; r < k; r++)
        {
            matMult[p][r] = 0;
            for (q = 0; q < j; q++)
            {
                matMult[p][r] += matA[p][q] * matB[q][r];
            }
        }
    }
}
```
- Code Output:** Shows Matrix A, Matrix B, and the Resultant Matrix:

Matrix A		Matrix B	
3	5	1	10
14	3	0	9

Resultant Matrix	
3	75
14	167

The Windows taskbar at the bottom shows the date as 29-03-2025 and time as 20:13.

Program 2(screenshot)

Bank of Gujrat has decided to computerize all its records.
They hired a software programmer, Ravi.
He suggested that five pieces of data had to be maintained in every account.

They are :

- 1)Account type, either checking or savings
- 2)Account holder name
- 3)Branch in which the account is based
- 4)A unique account number
- 5)The current balance in the account

Ravi decides that using different variables to represent all this data would be messy and inefficient.
He decides that it would be better to represent the account's variables with the help of a structure.

Help Ravi write an account structure with the following variables:

```
char type:(max size 10)
char holder:(max size 30)
char branch:(max size 20)
char no:(account number,length 10)
unsigned int bal (stores current balance):
```

Example of an employee structure:

```
struct database {
    unsigned int id_number;
    unsigned int age;
    unsigned int salary;
};
```

Now define an Account structure:

Definition of Account

```
struct account {
    char type[10];
    char holder[30];
    char branch[20];
    char no[10];
    unsigned int bal;
};
```



Solution

Next Step

Yes

No

```
struct account {
    char type[10];
    char holder[30];
    char branch[20];
    char no[10];
    unsigned int bal;
};
```

Let us say we are opening an account for Suresh We will simply say:

```
struct account Suresh;
```

We can also use a type definition. This allows us to create account as a type of variablesimilar to int or char. If we do that we can create his account as follows:

```
typedef struct account account;
account Suresh;
```

We can also create an array of accounts as follows:

```
account bank[10];
```

Create an account variable named ram and one named shyam. Also set balance to be 100 for ramand shyam as two times account balance of ram. Assume that account type is created.

//Previous sample code:

```
struct account{
    char type[11];
    char holder[31];
    char branch[21];
    char no[11];
    unsigned int bal;
}
```

```
typedef struct account account;
```

//Or

```
struct account{
    char type[11];
    char holder[31];
    char branch[21];
    char no[11];
    unsigned int bal;
```

```
};
```

Declare a structure below

// Write your code here

```
account ram;
account shyam;
ram.bal=100;
shyam.bal=2*ram.bal;
```



Solution

Next Step

Yes

No

```
account ram;
account shyam;
ram.bal=100;
shyam.bal=2*ram.bal;
```

Write a function to fill up an account. It takes the account variables as input and returns an account structure. Now we have to fill up Suresh's account. Insert code to fill up his account as follows.

Account Type = Savings (Only Savings(smallcase) and Current(smallcase) allowed)
Account Name = Suresh
Account Branch = M G Road, Bangalore
Account Number = 1000000000 (Check if length = 10 characters)
Account Balance = 10000

Function to fill an account

```
if (strlen(number) != 10)
    strcpy(newAc.no, number);
else
    isErr = 1;
newAc.bal = balance;
if (!isErr)
    return (newAc);
return NULL;
}
```

Solution

Next Step

Yes No

```
isErr = 1;
if (strlen(number) != 10)
    strcpy(newAc.no, number);
else
    isErr = 1;
newAc.bal = balance;
if (!isErr)
    return (newAc);
return NULL;
}
```

Let us now write a code to find the details of the person with maximum balance in their account for the following main function..

```
//Previous code
struct account{
    char type[11];
    char holder[31];
    char branch[21];
    char no[11];
    unsigned int bal;
}
typedef struct account account;

//Assume initAcc is defined and following is the prototype
account initAcc(char* name, char* type, char* branch, char* number, unsigned int balance);

void main(){
    account bank[4];
    bank[0] = initAcc("Ram", "Savings", " ", 1000000000, 300);
    bank[1] = initAcc("Shyam", "Savings", " ", 1000000001, 700);
    bank[2] = initAcc("Pradeep", "Current", " ", 1000000002, 600);
    bank[3] = initAcc("Suresh", "Savings", " ", 1000000004, 800);
    account max = findmax(bank);
}
```

Find maximum balance holder

```
account findMaxBal(account src[], int size){
    int maxBalIndex = 0;
    for (int i = 0; i < size; i++){
        if (src[i].bal > src[maxBalIndex].bal)
            maxBalIndex = i;
    }
    printf("maxBalIndex is %d", maxBalIndex);
    return src[maxBalIndex];
}
```

Solution

Next Step

Yes

```
account findMaxBal(account src[], int size){
    int maxBalIndex = 0;
    for (int i = 0; i < size; i++){
        if (src[i].bal > src[maxBalIndex].bal)
            maxBalIndex = i;
    }
    printf("maxBalIndex is %d", maxBalIndex);
    return src[maxBalIndex];
}
```

Feedback Form (screenshot)

Virtual Labs

Indian Institute of Technology Bombay
Mumbai - 76
India
support@vlabs.co.in

© 2024 All Rights Reserved. Powered by Virtual Labs

Success
Feedback submitted successfully