1. **Write a C program to read a one-dimensional array, print sum of all elements along with inputted array elements using Dynamic Memory Allocation.**

**Program**

#include<stdio.h>

#include<stdlib.h>

void main()

{

int i, n, \*arr, sum = 0;

printf("Enter the size of array\n");

scanf("%d", &n);

arr = (int \*)malloc(n\*sizeof(int));

if(arr == NULL)

{

printf("Memory not allocated\n");

exit(0);

}

printf("Enter Array Elements\n");

for(i=0; i<n; i++)

scanf("%d", &arr[i]);

printf("Entered Elements are\n");

for(i=0; i<n; i++)

{

sum=sum+arr[i];

printf("%d\n", arr[i]);

}

printf("The sum of array elements are: %d",sum);

}

1. **Write a C program to read and display text (string) using Dynamic Memory Allocation.**

**Program**

#include<stdio.h>

#include<stdlib.h>

void main()

{

int i, n;

char \*arr;

printf("Enter the size of string\n");

scanf("%d", &n);

arr = (char \*)malloc(n\*sizeof(char));

if(arr == NULL)

{

printf("Memory not allocated\n");

exit(0);

}

printf("Enter String\n");

scanf("%s", arr);

printf("Entered String is\n");

printf("%s", arr);

}

1. **Write a C program to read and display a two-dimensional array elements using Dynamic Memory Allocation.**

**Program**

#include<stdio.h>

#include<stdlib.h>

void main()

{

int r, c, i, j, \*\*arr;

printf("Enter rows and columns\n");

scanf("%d%d", &r,&c);

arr=(int \*\*)malloc(r\*sizeof(int \*));

if(arr == NULL)

{

printf("Memory not Allocated");

exit(0);

}

for(i=0; i<r; i++)

{

arr[i] = (int \*)malloc(c\*sizeof(int));

if(arr[i] == NULL)

{

printf("Memory not Allocated");

exit(0);

}

}

printf("Enter the values of the array\n");

for(i=0; i<r; i++)

{

for(j=0; j<c; j++)

{

scanf("%d", &arr[i][j]);

}

}

printf("Entered Elements are\n");

for(i=0; i<r; i++)

{

for(j=0; j<c; j++)

{

printf("%d\n", arr[i][j]);

}

}

for(i=0; i<r; i++)

free(arr[i]);

free(arr);

}

1. **Write a C program to read and print the details of only one student using structure and Dynamic Memory Allocation.**

**Program**

#include<stdio.h>

#include<stdlib.h>

void main()

{

struct student{

char name[20];

int rollno;

int age;

char course[20];

};

struct student \*stud;

stud = (struct student \*)malloc(sizeof(struct student));

if(stud == NULL)

{

printf("Memory not Allocated\n");

exit(0);

}

printf("Enter student details\n");

scanf("%s%d%d%s", stud->name, &stud->rollno, &stud->age, stud->course);

printf("Details of student are\n");

printf("%s\n%d\n%d\n%s\n", stud->name, stud->rollno, stud->age, stud->course);

}

1. **Write a C program to read and print the details of ‘n’ students using structure and Dynamic Memory Allocation.**

**Program**

#include<stdio.h>

#include<stdlib.h>

void main()

{

int n, i;

struct student{

char name[20];

int rollno;

int age;

char course[20];

};

struct student \*stud;

printf("Enter total number of students");

scanf("%d", &n);

stud = (struct student \*)malloc(n \* sizeof(struct student));

if(stud == NULL)

{

printf("Memory not Allocated\n");

exit(0);

}

for(i=0; i<n; i++)

{

printf("Enter Details of %d student",i+1);

scanf("%s%d%d%s", stud[i].name, &stud[i].rollno, &stud[i].age,

stud[i].course);

}

for(i=0; i<n; i++)

{

printf("Details of %d student are:\n",i+1);

printf("%s\n%d\n%d\n%s\n", stud[i].name, stud[i].rollno, stud[i].age,

stud[i].course);

}

}

1. **Create a structure to specify data on students given below: Roll number, Name, Department, Course, and Year of joining.**

**Assume that there are not more than 450 students in the collage.**

* 1. **(a) Write a function to print names of all students who joined in a particular year.**
  2. **(b) Write a function to print the data of a student whose roll number is given.**

**Program**

#include<stdio.h>

struct student{

int r\_n;

char name[20];

char dept[20];

char course[20];

int yoj;

}stud[450];

void display\_name(struct student [], int);

void display\_data(struct student [], int);

void main()

{

int i, n;

printf(“Enter the total number of students\n”);

scanf(“%d”, &n);

struct student stud[n];

for(i=0; i<n; i++)

{

printf(“Enter the roll number of %d student\n”, i+1);

scanf(“%d”, &stud[i].r\_n);

printf(“Enter the name of %d student\n”, i+1);

scanf(“%s”, stud[i].name);

printf(“Enter the department of %d student\n”, i+1);

scanf(“%s”, stud[i].dept);

printf(“Enter the course of %d student\n”, i+1);

scanf(“%s”, stud[i].course);

printf(“Enter the year of joining of %d student\n”, i+1);

scanf(“%d”, &stud[i].yoj);

}

display\_name(stud, n);

display\_data(stud, n);

}

void display\_name(struct student b[], int n)

{

int year, i;

printf(“Enter the year in which you want to see the names of all the students of that particular year\n”);

scanf(“%d”, &year);

for(i=0; i<n; i++)

{

if(b[i].yoj==year)

printf(“\n%s”, b[i].name);

}

}

void display\_data(struct student b[], int n)

{

int rn, i;

printf(“Enter the roll number of the student whose details you want to see\n”);

scanf(“%d”, &rn);

for(i=0; i<n; i++)

{

if(b[i].r\_n == rn)

{

printf(“%s\n”, b[i].name);

printf(“%s\n”, b[i].dept);

printf(“%s\n”, b[i].course);

printf(“%d\n”, b[i].yoj);

}

}

}

1. **Create a structure to specify data of customers in a bank. The data to be stored is: Account number, Name, Balance in account.**

**Assume maximum of 200 customers in the bank.**

* 1. **(a) Write a function to print the Account number and name of each customer with balance below Rs. 100.**
  2. **(b) If a customer request for withdrawal or deposit, it is given in the form: Acct. no, amount, code (1 for deposit, 0 for withdrawal)**

**Write a program to give a message, “The balance is insufficient for the specified withdrawal”.**

**Program**

#include<stdio.h>

struct bank{

char name[20];

int acc\_n;

int bal;

}cust[200];

int n;

void display\_name(struct bank [],int);

void withdraw(struct bank [],int);

void deposit(struct bank [],int);

void main()

{

int i,ch;

printf("Enter the total number of customers in a bank\n");

scanf("%d", &n);

struct bank cust[n];

for(i=0;i<n;i++)

{

printf("Enter the name of %d customer\n",i+1);

scanf("%s", cust[i].name);

printf("Enter the account number of %d customer\n",i+1);

scanf("%d", &cust[i].acc\_n);

printf("Enter the balance of %d customer\n",i+1);

scanf("%d", &cust[i].bal);

}

while(1)

{

printf("Enter your choice\n");

printf("0. To Withdraw money from a particular account\n");

printf("1. To Deposit money in a particular account\n");

printf("2. To get the names of all those customers with balance less than Rs. 100\n");

printf("3. Exit\n");

scanf("%d", &ch);

switch(ch)

{

case 0: withdraw(cust,n);

break;

case 1: deposit(cust,n);

break;

case 2: display\_name(cust,n);

break;

case 3: exit(0);

default: printf("Please Enter a valid option\n");

}

}

}

void display\_name(struct bank cust[], int n)

{

int i, count=0;

for(i=0;i<n;i++)

{

if(cust[i].bal<=100)

{

printf("%s\n", cust[i].name);

printf("%d\n",cust[i].acc\_n);

count++;

}

}

if(count==0)

printf("No Customer has balance less than 100\n");

}

void withdraw(struct bank cust[], int n)

{

int an, i, amount;

printf("Enter the Account number\n");

scanf("%d", &an);

for(i=0;i<n;i++)

{

if((cust[i].acc\_n)==an)

{

if((cust[i].bal)<=100)

{

printf("The balance is insufficient for the specified withdrawal\n");

break;

}

else

{

printf("Enter the Amount to withdraw\n");

scanf("%d", &amount);

cust[i].bal=cust[i].bal-amount;

printf("Collect your money from the cash counter");

}

}

}

}

void deposit(struct bank cust[], int n)

{

int amount,account,i;

printf("Enter the account number and amount\n");

scanf("%d%d",&account, &amount);

for(i=0;i<n;i++)

{

if(cust[i].acc\_n==account)

cust[i].bal=cust[i].bal+amount;

}

printf("Amount Deposited\n");

}

1. **There is a structure called employee that holds information like employee code, name, date of joining. Write a program to create an array of the structure and enter some data into it. Then ask the user to enter current date. Display the names of those employees whose tenure is 3 or more than 3 years according to the given current date.**

**Program**

#include<stdio.h>

struct date{

int day, month, year;

};

int check\_date(struct date \*);

void main()

{

int n,i, chkdt;

struct date curr;

struct employee

{

int code;

char emp\_name[20];

struct date doj;

};

printf("Enter the total number of employees\n");

scanf("%d", &n);

struct employee emp[n];

printf("Enter the information for %d employees\n",n);

for(i=0; i<n; i++)

{

printf("Employee Name\n");

scanf("%s",emp[i].emp\_name);

printf("Employee Code\n");

scanf("%d",&emp[i].code);

printf("Date of Joining\n");

chkdt=check\_date(&emp[i].doj);

if(chkdt == 0)

{

printf("Improper Date\n");

exit(0);

}

}

printf("Enter current date\n");

chkdt=check\_date(&curr);

if(chkdt == 0)

{

printf("Improper date entered\n");

exit(0);

}

//Printing the names of employees whose tenure is more than 3 years

for(i=0;i<n;i++)

{

if(curr.year>emp[i].doj.year+3)

{

printf("Name of the employee who has completed 3 years is:\n%s", emp[i].emp\_name);

break;

}

else

{

if(curr.year == emp[i].doj.year+3)

{

if(curr.month>emp[i].doj.month)

{

printf("Name of the employee who has completed 3 years is:\n%s",emp[i].emp\_name);

}

else

{

if((curr.month == emp[i].doj.month) && (curr.day >= emp[i].doj.day))

printf("Name of the employee who has completed 3 years is:\n%s",emp[i].emp\_name);

}

}

}

}

}

int check\_date(struct date \*dt)

{

printf("Enter day\n");

scanf("%d",&dt->day);

printf("Enter month\n");

scanf("%d",&dt->month);

printf("Enter year\n");

scanf("%d",&dt->year);

if((dt->day>31 || dt->day<0) ||

(dt->month>12 || dt->month <0) ||

(dt->year>9999 || dt->year <1000))

return 0;

else

return 1;

}

1. **Write a program that compares two given dates. To store date use structure say date that contains three members namely date, month and year. If the dates are equal then display message as "Equal" otherwise "Unequal".**

**Program**

#include<stdio.h>

struct date{

int day, month, year;

};

int check\_date(struct date \*);

void main()

{

int chkdt;

struct date d1, d2;

printf("Enter the dates to be compared\n");

chkdt=check\_date(&d1);

if(chkdt == 0)

exit(0);

chkdt=check\_date(&d2);

if(chkdt == 0)

exit(0);

//Comparing two dates

if((d1.day == d2.day) && (d1.month == d2.month) && (d1.year == d2.year))

printf("Dates are equal\n");

else

printf("Dates are not equal\n");

}

int check\_date(struct date \*dt)

{

printf("Enter day\n");

scanf("%d",&dt->day);

printf("Enter month\n");

scanf("%d",&dt->month);

printf("Enter year\n");

scanf("%d",&dt->year);

if((dt->day>31 || dt->day<0) ||

(dt->month>12 || dt->month <0) ||

(dt->year>9999 || dt->year <1000))

return 0;

else

return 1;

}

1. **Write a menu driven program that depicts the working of a library. The menu options should be:** 
   1. **1. Add book information**
   2. **2. Display book information**
   3. **3. List all books of given author**
   4. **4. List the title of specified book**
   5. **5. List the count of books in the library**
   6. **6. List the books in the order of accession number**
   7. **7. Exit**

**Create a structure called library to hold accession number, title of the book, author name, price of the book, and flag indicating whether book is issued or not.**

**Practice this question on the basis of the solutions provided for the first 4 questions.**