

Write a program to calculate the gross salary of Ramesh if the dearness allowance is 40% of the salary and house rent allowance is 20% of the salary.

```
#include <stdio.h>
int main()
{
    int dearness_allowance_percent = 40;
    int house_rent_allowance_percent = 20;
    float salary;
    float gross_salary;

    printf ("Enter the salary of Ramesh:");
    scanf ("%f", &salary);

    gross_salary = salary + (((float)dearness_allowance_percent / 100) * salary) + (((float)house_rent_allowance_percent / 100) * salary);

    printf ("the gross salary of Ramesh is %.f", gross_salary);
    return 0;
}
```

If the marks obtained by student in 5 subjects is given through the keyboard then write a program to find the total marks and percentage assuming the full marks of each subject is 100.

```
#include <stdio.h>

int main()
{
    float sub1, sub2, sub3, sub4, sub5;
    float percentage, total_marks;

    printf ("Enter the marks in 5 subjects:");
    scanf ("%f %f %f %f %f", &sub1, &sub2, &sub3,
           &sub4, &sub5);

    total_marks = sub1 + sub2 + sub3 + sub4 + sub5;
    percentage = (total_marks / 500) * 100;

    printf ("the total marks obtained is %f\n", total_marks);
    printf ("the percentage obtained is %f", percentage);
```

If a five digit number is input through the keyboard, write a program to calculate the sum of its digits.

```
#include <stdio.h>

int main()
{
    int num;
    int sum=0;
    printf("enter a 5 digit number");
    scanf ("%d", &num)

    while (num!=0)
    {
        sum= sum + (num % 10);
        num= num / 10;
    }

    printf ("the sum of its digit is %d", sum);
    return 0;
}
```

consider a currency system in which there are notes of Re 1, Rs 2, Rs 5, Rs 10, Rs 20, Rs 50, Rs 100, Rs 500 and Rs 1000. If the sum of Rs N is entered through the keyboard, write a program to compute the smallest number of notes that will combine to give Rs N.

```
#include <stdio.h>
int main()
{
    int N;
    int count1, count2, count5, count10, count20, count50,
        count100, count500, count1000;

    count1 = count2 = count5 = count10 = count20 = count50 =
    count100 = count500 = count1000 = 0;

    printf("enter the sum of money");
    scanf ("%d", &N);

    while (N != 0)
    {
        if (N >= 1000)
        {
            count1000++;
            N = N - 1000;
            continue;
        }
        if (N >= 500)
        {
            count500++;
            N = N - 500;
            continue;
        }
        if (N >= 100)
        {
            count100++;
            N = N - 100;
            continue;
        }
        if (N >= 50)
        {
            count50++;
            N = N - 50;
            continue;
        }
        if (N >= 20)
        {
            count20++;
            N = N - 20;
            continue;
        }
    }
}
```

```
if (N>=10){  
    count10++;  
    N-=10;  
    continue;  
}  
if (N>=5){  
    count5++;  
    N-=5;  
    continue;  
}  
if (N>=2){  
    count2++;  
    N-=2;  
    continue;  
}  
if (N==1){  
    count1++;  
    N-=1;  
    continue;  
}
```

```
printf("the number of notes required is\n %.d of RS  
1000\n %.d of RS 500\n %.d of RS 100\n %.d of  
RS 50\n %.d of RS 20\n %.d of RS 10\n %.d of  
RS 5\n %.d of RS 2 and\n %.d of Re 1", count1000,  
count500, count100, count50, count20, count10,  
count5, count2, count1);
```

```
return 0;
```

```
}
```

Q) According to the Gregorian calendar, it was Monday on the date 2001/01/01. If any year is input through the keyboard, write a program to find what is the day on January 1st of that year.

```
#include <stdio.h>
int main()
{
    int day;
    int year;
    int additional_days;
    int days_to_add;
    printf("enter the year you want to find the day of
           January 1 ");
    scanf("%d", &year);
    days_to_add = (year - 2001) + ((year - 2001) / 4);
    additional_days = days_to_add % 7;
    day = additional_days + 2
    switch (day)
    {
        case 1:
            printf("the day is Sunday");
            break;
        case 2:
            printf("the day is Monday");
            break;
        case 3:
            printf("the day is Tuesday");
            break;
        case 4:
            printf("the day is Wednesday");
            break;
        case 5:
            printf("the day is Thursday");
            break;
        case 6:
            printf("the day is Friday");
            break;
    }
}
```

default:

```
printf("the day is Saturday");
```

}

Q) Given the co-ordinates (x, y) of center of a circle and its radius, write a program to find whether a point lies inside the circle, on the circle or outside the circle.

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

```
#include <math.h>
```

```
int main()
```

{

```
float r, h, k, x, y;
```

```
float d=0;
```

```
printf("Enter the point of center of the circle");
scanf("%f %f", &h, &k);
```

```
printf("Enter the radius of circle");
scanf("%f", &r);
```

```
printf("Enter the point to be checked");
scanf("%f %f", &x, &y);
```

```
d = sqrt(pow((x-h), 2) + pow((y-k), 2));
```

```
if(d>r)
```

```
    printf("the point lies outside the circle");
```

```
if else (d < r)
```

```
    printf("the point lies inside the circle");
```

```
else
```

```
    printf("the point lies on the circle");
```

}

Q) Write a program to take any character as input from the user and check if it is upper case, lower case, a digit or a special character.

```
#include <stdio.h>
int main()
{
    char c=0;
    printf ("enter a character: ");
    scanf ("%c", &c);

    if (c>= 65 && c<=90)
        printf ("It is a capital letter");
    else if (c>=97 && c<=122)
        printf ("It is a small letter");
    else if (c>=48 && c<=57)
        printf ("It is a digit");
    else
        printf ("It is a special character");
}
```

Q) Write a program that converts RGB color to CMYK color.

```
#include <stdio.h>
int main()
{
    float r,g,b;
    float w,c,m,y,k;
    printf ("enter the red, green and blue value of
            RGB format");
    scanf ("%f %f %f", &r, &g, &b);
    w = ((r/255) >= (g/255) && (r/255) >= (b/255)) ?
        r/255 : (((g/255) >= (r/255) && (g/255) >= b/255))?
        g/255 : b/255;

    c = (w - r/255)/w;
    m = (w - g/255)/w;
    y = (w - b/255)/w;
    k = 1-w;
    printf ("the CMYK format is %f %f %f %f",
            c, m, y, k);
}
```

3

Q.) Write a program to print all armstrong numbers between 1 and 500.

```
#include <stdio.h>
#include <math.h>
int main()
{
    int n=1;
    int a=0;
    printf("%d", n);
    int num;
    while (n <= 500)
    {
        a=n;
        x=0;
        while (a!=0)
        {
            num=a%10;
            x=x+(num*num*num);
            a=a/10;
        }
        if (x==n)
            printf("%d\n", n);
        n++;
    }
    return 0;
}
```

Q) Write a program that receives decimal and prints its octal equivalent.

```
#include <stdio.h>
int main()
{
    int dec;
    printf("enter the decimal number");
    scanf("%d", &dec);
    int oct=0;
    while (dec != 0)
    {
        oct = (oct * 10) + (dec % 8);
        dec = dec / 8;
    }
    int oct1=0;
    while (oct != 0)
    {
        oct1 = (oct * 10) + (oct % 10);
        oct = oct / 10;
    }
    printf("octal equivalent is %d", oct);
```

Q.7 Write a program to print all prime numbers from 1 to 300.

```
#include <stdio.h>
int main()
{
    int n=1;
    while (n<=300)
    {
        int c=2;
        while (c<n)
        {
            if (n%c == 0)
                break;
            c++;
        }
        if (c==n)
            printf("%d \n",n);
        n++;
    }
}
```

Q.8) write a program to print the multiplication table of the number entered by the user.

```
#include <stdio.h>
int main()
{
    int n;
    printf("enter the number you want to find the
           multiplication table of: ");
    scanf ("%d", &n);
    for (int i=1; i<=11; i++)
        printf ("%d * %d = %d \n", n, i, (n*i));
}
```

Q> A 5 digit positive integer is entered through the keyboard, write a recursive function to calculate sum of digits.

```
#include <stdio.h>
int recur(int);
int main()
{
    int x;
    printf("enter a five digit number:");
    scanf("%d", &x);
    printf("the sum of the digits is %d", recur(x));
    return 0;
}

int recur(int a)
{
    int sum;
    if (a == 0)
        return 0;
    else
        sum = recur(a/10) + (a%10);
    return sum;
}
```

Q.) A positive integer is entered through the keyboard, write a program to obtain the prime factorial of a number.

```
#include <stdio.h>
void prime_fac(int);
int check_prime(int);
int main()
{
    int sc;
    printf("Enter the number you want to find the prime
factorial of.");
    scanf("%d", &sc);
    prime_fac(sc)
}

void prime_fac(int a)
{
    while (a != 1) {
        for (int i = 2; i < a; i++) {
            if ((check_prime(i) == 1) && ((a % i) == 0)) {
                printf("\n%d\t", i);
                a = a / i;
                break;
            }
        }
        if ((check_prime(a) == 1)) {
            printf("\n%d", a);
            a = 1;
        }
    }
    continue;
}

int check_prime(int num) {
    int prime = 1;
    int a = 2;
    for (a; a < num; a++) {
        if (num % a == 0) {
            prime = 0;
            break;
        }
    }
    return prime;
}
```

Q) Write a menu driven program which has following options

- 1) Factorial of a number
- 2) Prime or not
- 3) Odd or Even
- 4) Exit

```
#include <stdio.h>
int main()
{
    int choice;
    start:
    printf("\n Choose what you want to perform\n 1. Factorial of a number\n 2. Prime or not\n 3. Odd or Even\n 4. Exit\n");
    printf("Enter your choice");
    scanf("%d", &choice);
    int num;
    switch(choice)
    {
        case 1:
            printf("enter the number you want to find the factorial of");
            scanf("%d", &num);
            int fac=1;
            while (num!=0)
            {
                fac=fac*num;
                num--;
            }
            printf("The factorial is %d", fac);
            goto start;
        case 2:
            printf("enter the number you want to check if it is prime or not");
            scanf("%d", &num);
            int n=2;
            while (n!=num)
            {
                if (num%n==0)
                {
                    printf("The number is not prime");
                    break;
                }
                n++;
            }
    }
}
```

```
if (n==num)
    printf("the number is a prime number");
    goto start;

case 3:
    printf("Enter the number that you want to check
        if it is odd or even");
    scanf("%d", &num);
    if (num % 2 != 0)
        printf("It is an odd number");
    else
        printf("It is an even number");
    goto start;

case 4:
    break;
}
```

Q) Write a function that receive 5 integers and return the sum, average and standard deviation of these numbers. call the function from the main() and print the result in the main().

```
#include <stdio.h>
#include <math.h>
int program(float*, float*, float*);
int main()
{
    float sum, average, standard_deviations;
    program(&sum, &average, &standard_deviations);
    printf("the sum of number is %.f\n", sum);
    printf("the average of numbers is %.f\n", average);
    printf("the standard deviation is %.f\n",
        standard_deviations);
}
```

```
int program(float *s, float*a, float*sd)
{
    float b, c, d, e, f;
    printf("enter 5 numbers: ");
    scanf(" %f %f %f %f %f", &b, &c, &d, &e, &f);
    *s = b+c+d+e+f;
    *a = *s/5;
    *sd = sqrt(((pow(f-*a, 2))+(pow((b-*a), 2))+
        (pow((c-*a), 2))+(pow((d-*a), 2))+(pow((e-*a), 2))
    ))/5);
```

3

Q) Write a function
of a raised to b.
 $\text{ex}(a, b)$ to calculate the value

```
#include <stdio.h>
int ex(int, int);
int main()
{
    int x, a;
    printf("enter the number in coefficient.");
    scanf("%d", &a);
    printf("enter the number in power");
    scanf("%d", &b);
    printf("the result is %d", ex(a, b));
    return 0;
}

int ex(int a, int b)
{
    int ans = 1;
    while (b != 0)
    {
        ans = ans * a;
        b--;
    }
    return ans;
}
```

g) Any year entered through the keyboard. Write a function to determine whether the year is a leap year or not.

```
#include <stdio.h>
void check (int);
int main()
{
    int year;
    printf ("enter the year");
    scanf ("%d", &year);
    check (year);
}

void check (int y)
{
    if ((y % 4 == 0 && y % 100 != 0) || (y % 100 == 0
        && y % 400 == 0))
        printf ("the year is a leap year");
    else
        printf ("the year is not a leap year");
}
```

Q) Write a function to compute the greatest common divisor using Euclid's algorithm.

```

#include <stdio.h>
int divisor(int, int);
int main()
{
    int x;
    int y;
    printf("enter two numbers you want to find the common divisor of (please put the greater number first)");
    scanf("%d %d", &x, &y);
    printf("the highest common divisor is %d",
    divisor(x, y));
}

// EUCLID'S ALGORITHM.
int divisor(int first, int second)
{
    int quotient;
    quotient = first / second;
    int answer = second;
    int temp = second;
    second = first - (quotient * second);
    while (second != 0)
    {
        quotient = temp / second;
        answer = second;
        second = temp - (quotient * second);
        temp = answer;
    }
    return answer;
}

```

Q) Write a program that takes random numbers as input from the user and arranges them in ascending order.

```
#include <stdio.h>
int main()
{
    int num;
    printf ("what is the number of inputs");
    scanf ("%d", &num);
    int a[num];
    int n;
    int o;
    int j;
    printf ("enter the input numbers");
    for (int i=0; i<num; i++)
        scanf ("%d", &a[i]);
    for (int i=0; i<num; i++)
    {
        int min=a[i];
        o=a[i];
        for (j=1; j<num; j++)
        {
            if (a[j]<min)
            {
                min=a[j];
                n=j;
            }
        }
        a[i]=min;
        if (min!=o)
            a[n]=o;
    }
    printf ("the arranged order is:");
    for (int i=0; i<num; i++)
    {
        printf ("%d/t", a[i]);
    }
}
```

Q) Write a program that extracts parts of the given string from the desired location.

```
#include <stdio.h>
int main () {
    char string[20];
    printf ("enter the string:");
    scanf ("%[^\\n]s", string);
    int f, e;
    printf ("enter the position and number of characters
            you want to print");
    scanf ("%d %d", &f, &e);
    if (e != 0) {
        int j = 0;
        for (int i = f; j < e; i++) {
            printf ("%c", string[i]);
            j++;
        }
    } else
    {
        int i = f;
        while (string[i] != '\0') {
            printf ("%c", string[i]);
            i++;
        }
    }
}
```

Q) Write a program to multiply two 3×3 matrices.

```
#include <stdio.h>
int main() {
    int matrixa[3][3];
    int matrixb[3][3];
    printf("Enter the elements of matrix A");
    for (int i=0; i<3; i++)
        for (int j=0; j<3; j++)
            scanf("%d", &matrixa[i][j]);
    printf("Enter the elements of matrix B");
    for (int i=0; i<3; i++)
        for (int j=0; j<3; j++)
            scanf("%d", &matrixb[i][j]);
    int result_matrix[3][3];
    for (int i=0; i<3; i++)
        for (int j=0; j<3; j++) {
            int sum=0;
            for (int k=0; k<3; k++)
                sum = sum + (matrixa[i][k] * matrixb[k][j]);
            printf("%d ", sum);
        }
    printf("\n");
}
```

Q) Write a program to find if a square matrix is symmetric.

```
#include <stdio.h>
int main()
{
    int matrix[3][3];
    printf("Enter the elements of the matrix:");
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            scanf("%d", &matrix[i][j]);
        }
    }

    int tmatrix[3][3];
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            tmatrix[i][j] = matrix[j][i];
        }
    }

    int flag = 1;
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            if(matrix[i][j] != tmatrix[i][j])
                flag = 0;
        }
    }

    if(flag == 1)
        printf("It is symmetric");
    else
        printf("It is not symmetric");
}
```

Q) Write a program to copy the contents of one array into another in the reverse order.

```
#include <stdio.h>
int main()
{
    int n;
    printf("enter the number of elements of an array.");
    scanf("%d", &n);
    int arr[n];
    int reverse[n];
    printf("enter the elements of an array.");
    for (int i=0; i<n; i++) {
        scanf("%d", &arr[i]);
    }

    for (int i=(n-1), j=0; i>=0, i--) {
        reverse[i] = arr[j];
        j++;
    }

    printf ("the elements of the reversed array are.");
    for (int i=0; i<n; i++) {
        printf ("%d\t", reverse[i]);
    }
}
```

Create a structure to specify data on students given below:

Roll no, Name, Department, course, Year of joining.
Assume that there are not more than 450 students in the college.

1) Write a function to print names of all students who joined in a particular year.

2) Write a function to print the data of student whose roll no is received.

```
#include <stdio.h>
struct student {
    int rollno;
    char name[20];
    char course[20];
    char department[40];
    int year_of_joining;
};

void yearchoice(struct student stu[]);
void rollnocheck(struct student st[]);
int main() {
    struct student stud[450];
    char action='y';
    int i=0;
    while (action != 'n' && i<450) {
        printf("for the student in position %d\n", i+1);
        printf("enter your name");
        gets(stud[i].name);
        fflush(stdin);
        printf("enter the roll no");
        scanf("%d", &stud[i].rollno);
        fflush(stdin);
        printf("enter the department: ");
        gets(stud[i].department);
        fflush(stdin);
        printf("enter the course: ");
        gets(stud[i].course);
        fflush(stdin);
        printf("enter the year of joining");
        scanf("%d", &stud[i].year_of_joining);
        fflush(stdin);
    }
}
```

```

printf("Do you want to enter more data?(y/n)");  

scanf("%c", &action);  

fflush(stdin);  

i++;  

printf("\n");
}
printf("\n");
yearchoice(stud);
printf("\n");
rollnocheck(stud);
}

void yearchoice(struct student stud[])
{
int year-of-choice;
printf("enter the year you want to check the  

list of students:");
scanf("%d", &year-of-choice);
int j=1;
for(int i=0; i<450; i++)
{
if(stu[i].year-of-joining == year-of-choice)
{
printf("%d.%s\n", j, stu[i].name);
j++;
}
}
}

void rollnocheck(struct student st[])
{
int checkrn;
printf("enter the roll no of student: ");
scanf("%d", &checkrn);
for(int i=0; i<450; i++)
{
if(checkrn == st[i].rollno)
{
printf("Name: %s\n", st[i].name);
printf("Department: %s\n", st[i].department);
printf("course: %s\n", st[i].course);
printf("Year of joining: %d", st[i].year-of-joining);
}
}
}

```

- Q) 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 customer in the bank.
- 1) Write a function to print the Account number and name of each customer with balance below RS 100.
 - 2) If a customer requests for withdrawl or deposit, the form contains the field; Acc. No, amount, code C for deposit, 0 for withdrawl). Write a program to give a message, "The balance is insufficient for the specified withdrawl". If on withdrawl the balance is below RS 100.

```
#include<stdio.h>
struct account_details
{
    double account_number;
    char name[30];
    float balance;
};

void below100(struct account_details accdet[]);
void transaction(struct account_details accdet[]);

int main()
{
    struct account_details ad[200];
    int set=0;
    while (set < 200)
    {
        ad[set].account_number=0;
        set++;
    }

    char action='y';
    int i=0;
    while (action != 'n' && i < 200)
    {
        printf("Enter the account no. ");
        scanf("%d", &ad[i].account_number);
        fflush(stdin);
        printf("Enter the name: ");
        gets(ad[i].name);
        fflush(stdin);
        printf("Enter the bank balance: ");
        printf("Enter the bank balance: ");
        scanf("%f", &ad[i].balance);
        fflush(stdin);
        printf("Do you have to enter more data? ");
        scanf("y/n", &action);
    }
}
```

```

    i++;
    printf("\n");
}
int choice;
top:
printf("\n What do you want to perform next?\n 1.
whose balance and account number of customer
transaction\n 3. Exit\n");  

scanf("y-d", &choice);
printf("\n");
switch(choice){
case 1:
    below_100(acd);
    goto top;
case 2:
    transaction(acd);
    goto top;
case 3:
    break;
}
void below_100(struct account_details accdet[]){
int n=0;
int j=1;
while (n<200){
    if ((accdet[n].balance<100 && accdet[n].
        account_number!=0)) {
        printf("Y-d. Name %s\t", j, accdet[n].name);
        printf("Account Number: %.0lf", accdet[n].
            account_number);
        printf("\n");
        j++;
    }
}
n++;
}
void transaction(struct account_details accdet[]){
int choice;
printf("What do you want to perform?\n Enter 1
for deposit and 0 for withdrawl. ");
scanf("y-d", &choice);
double AccNO;
float am;
}

```

```

switch (choice) {
    case 0:
        printf("Enter Account Number: ");
        scanf("%d", &AccNo);
        printf("Enter the amount to withdraw: ");
        int n=0;
        while(n<200)
    {
        if (acc-det[n].account_number == AccNo)
            break;
    }
    if ((acc-det[n].balance-am)<100)
        printf("The balance is insufficient for the
    else
        printf("Your transaction was successful\n
    Your remaining balance is %.2f\n", acc-det[acc-det[n].balance=acc-det[n].balance-am];
    break;
}

case 1:
    printf("Enter your Account Number: ");
    scanf("%d", &AccNo);
    printf("Enter the amount you want to deposit");
    scanf("%d", &am);
    int m=0
    while(m<200)
    {
        if (acc-det[m].account_number == AccNo)
            break;
    }
    printf("Your transaction was successful.\n
    Your new balance is %.2f.\n", acc-det[m].balance+am);
    acc-det[m].balance+=am;
    break;
}

```

Q) Program to calculate the sum of n numbers entered by the user.

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    int n, i, *ptr, sum=0;
    printf ("Enter number of elements");
    scanf ("%d", &n);
    ptr = (int *) malloc (n * sizeof (int));
    if (ptr == NULL) {
        printf ("Memory not allocated");
        exit (1);
    }
    printf ("Enter elements:");
    for (i=0; i<n; ++i) {
        scanf ("%d", ptr+i);
        sum += *(ptr+i);
    }
    printf ("sum = %d", sum);
    free (ptr);
    return 0;
}
```

Q) program demonstration of `calloc` and `realloc`.

```

#include <stdio.h>
#include <stdlib.h>
int main() {
    int *p, i, n;
    printf("Initial size of the array is 4\n");
    p = (int *)calloc(4, sizeof(int));
    if (p == NULL) {
        printf("Memory allocation failed");
        exit(1);
    }
    for (i = 0; i < 4; i++) {
        printf("Enter elements at index %d: ", i);
        scanf("%d", p + i);
    }
    printf("\nIncreasing size of array to 5");
    p = (int *)realloc(p, 5 * sizeof(int));
    if (p == NULL) {
        printf("Memory allocation failed");
        exit(1);
    }
    printf("Enter 1 more element");
    scanf("%d", p + 4);
    printf("Final Array\n");
    for (int i = 0; i < 5; i++)
        printf("%d", *(p + i));
    return 0;
}

```

Q7 Write a C program to read name and marks of n number of students from and store them in a file. If the file previously exists, add the information to the file.

```
#include <stdio.h>
int main()
{
    char name[50];
    int marks, i, num;
    printf("Enter number of students: ");
    scanf("%d", &num);
    FILE *fptr;
    fptr = fopen("D:/student.txt", "a");
    if (fptr == NULL)
    {
        printf("Error!");
        exit(1);
    }
    for (i=0; i<num; ++i)
    {
        printf("For student %d\n Enter name: ", i+1);
        scanf("%s", name);
        printf("Enter marks: ");
        scanf("%d", &marks);
        fprintf(fptr, "\n Name: %s\n Marks= %d\n", name,
                marks);
    }
    fclose(fptr);
    return 0;
}
```

Q) Write a program to write all the members of an array of structures to a file using fwrite(). Read the array from the file and display it on the screen.

```
#include <stdio.h>
struct student
{
    char name[50];
    int height;
};

int main()
{
    struct student stud1[5], stud2[5];
    FILE *fptr;
    int i;
    fptr = fopen("file.txt", "wb");
    for (i=0; i<5; ++i)
    {
        fflush(stdin);
        printf("Enter name: ");
        gets(stud1[i].name);
        printf("Enter height: ");
        scanf("%d", &stud1[i].height);
    }
    fwrite(stud1, sizeof(stud1), 1, fptr);
    fclose(fptr);

    fptr = fopen("file.txt", "rb");
    fread(stud2, sizeof(stud2), 1, fptr);
    for (i=0; i<5; ++i)
    {
        printf("Name: %s\nHeight: %d", stud2[i].name, stud2[i].height);
    }
    fclose(fptr);
}
```

3

Q) program to draw a line.

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

void main(void)
{
    int gd=DETECT, gmode;
    int x1=200, y1=200;
    int x2=300, y2=300;
    clrscr();
    initgraph(&gd, &gmode, "c:\tc\bg1");
    line(x1, y1, x2, y2);
    getch();
    closegraph();
}
```

Q) program for drawing a circle.

```
#include <graphics.h>
int main()
{
    int gd=DETECT, gm;
    initgraph(&gd, &gm, "c:\tc\bg1");
    circle(250, 200, 50);
    getch();
    closegraph();
    return 0;
}
```