**C PROGRAMS**

**LAB SESSSION 4**

**HAL PYRAMID OF \***

#include <stdio.h>

int main() {

int i, j, rows;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = 1; i <= rows; ++i) {

for (j = 1; j <= i; ++j) {

printf("\* ");

}

printf("\n");

}

return 0;

}

**Half Pyramid of Numbers**

#include <stdio.h>

int main() {

int i, j, rows;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = 1; i <= rows; ++i) {

for (j = 1; j <= i; ++j) {

printf("%d ", j);

}

printf("\n");

}

return 0;

}

### Half Pyramid of Alphabets

#include <stdio.h>

int main() {

int i, j;

char input, alphabet = 'A';

printf("Enter an uppercase character you want to print in the last row: ");

scanf("%c", &input);

for (i = 1; i <= (input - 'A' + 1); ++i) {

for (j = 1; j <= i; ++j) {

printf("%c ", alphabet);

}

++alphabet;

printf("\n");

}

return 0;

}

### Inverted half pyramid of \*

#include <stdio.h>

int main() {

int i, j, rows;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = rows; i >= 1; --i) {

for (j = 1; j <= i; ++j) {

printf("\* ");

}

printf("\n");

}

return 0;

}

### Inverted half pyramid of numbers

#include <stdio.h>

int main() {

int i, j, rows;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = rows; i >= 1; --i) {

for (j = 1; j <= i; ++j) {

printf("%d ", j);

}

printf("\n");

}

return 0;

}

### Full Pyramid of \*

#include <stdio.h>

int main() {

int i, space, rows, k = 0;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = 1; i <= rows; ++i, k = 0) {

for (space = 1; space <= rows - i; ++space) {

printf(" ");

}

while (k != 2 \* i - 1) {

printf("\* ");

++k;

}

printf("\n");

}

return 0;

}

### Full Pyramid of Numbers

#include <stdio.h>

int main() {

int i, space, rows, k = 0, count = 0, count1 = 0;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = 1; i <= rows; ++i) {

for (space = 1; space <= rows - i; ++space) {

printf(" ");

++count;

}

while (k != 2 \* i - 1) {

if (count <= rows - 1) {

printf("%d ", i + k);

++count;

} else {

++count1;

printf("%d ", (i + k - 2 \* count1));

}

++k;

}

count1 = count = k = 0;

printf("\n");

}

return 0;

}

### Inverted full pyramid of \*

#include <stdio.h>

int main() {

int rows, i, j, space;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = rows; i >= 1; --i) {

for (space = 0; space < rows - i; ++space)

printf(" ");

for (j = i; j <= 2 \* i - 1; ++j)

printf("\* ");

for (j = 0; j < i - 1; ++j)

printf("\* ");

printf("\n");

}

return 0;

}

### Pascal's Triangle

#include <stdio.h>

int main() {

int rows, coef = 1, space, i, j;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = 0; i < rows; i++) {

for (space = 1; space <= rows - i; space++)

printf(" ");

for (j = 0; j <= i; j++) {

if (j == 0 || i == 0)

coef = 1;

else

coef = coef \* (i - j + 1) / j;

printf("%4d", coef);

}

printf("\n");

}

return 0;

}

### Floyd's Triangle

#include <stdio.h>

int main() {

int rows, i, j, number = 1;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = 1; i <= rows; i++) {

for (j = 1; j <= i; ++j) {

printf("%d ", number);

++number;

}

printf("\n");

}

return 0;

}

### LAB SESSION 5

### STRUCTURE PROGRAMS

### #include<stdio.h>

### struct worker

### {

### char name[20];

### int wage;

### int wdays;

### };

### int main()

### {

### struct worker a,b;

### printf("Enter details of First Worker\n");

### printf("-----------------------\n");

### printf("Enter first worker name: ");

### scanf("%s",a.name);

### printf("Enter wage: ");

### scanf("%d",&a.wage);

### printf("Enter Wdays: ");

### scanf("%d",&a.wdays);

### printf("------------------------\n");

### printf("Enter details of second worker\n");

### printf("-----------------------\n");

### printf("Enter second worker name:");

### scanf("%s",b.name);

### printf("Enter wage: ");

### scanf("%d",&b.wage);

### printf("Enter wdays: ");

### scanf("%d",&b.wdays);

### printf("----------------------\n");

### int p1=a.wage\*a.wdays;

### printf("Name of the first worker:%s\nPayment of the first worker:%d\n",a.name,p1);

### printf("--------------------------------------------\n");

### int p2=b.wage\*b.wdays;

### printf("Name of the second worker:%s\nPayment of the second worker:%d\n",b.name,p2);

### printf("--------------------------------------------\n");

### return 0;

### }