----------------------------------------------------programs on const-------------------------------------------------------------  
  
1

#include<stdio.h>

int main(){

    const int arr[]={10,20,30,40,50};

    for(int i=0;i<5;i++){

        printf("The element in the %dth pos =%d\n",i,arr[i]);

        arr[0]=22;

    }

    return 0;

}

2

#include<stdio.h>

int main(){

    const int last=6;

    int count=0;

    for(int i=0;i<=last;i++){

        count+=1;

    }

    printf("the count is %d",count);

}

3

#include<stdio.h>

int main(){

    const int rows=5;

    const int columns=5;

    int count=0;

    for(int i=0;i<rows;i++){

        for(int j=0;j<rows;j++){

            count+=1;

        }

    }

    printf("total count =%d",count);

}

4

#include<stdio.h>

int main(){

    int arr[]={4,5,6,7,8};

    int \*const ptr=arr;

    int size=sizeof(arr)/sizeof(arr[0]);

    for(int i=0;i<size;i++){

        printf("Value at arr[%d]: %d\n", i, \*(ptr + i));

    }

    return 0;

}

5

#include<stdio.h>

int main(){

    const int pi=3.14;

    int n,total\_area=0,area=0;

    printf("enter the radius of starting circle");

    scanf("%d",&n);

    for(int i=0;i<n;i++){

        area=pi\*i\*i;

    }

    total\_area+=area;

    printf("total area=%d",area);

}

6

#include<stdio.h>

int main(){

    int count=0;

    const int termination=10;

    while(count<=termination){

        printf("the value in the termination=%d\n",count);

        count+=1;

    }

    return 0;

}

7

#include <stdio.h>

int main() {

    const int STEP\_SIZE = 3;

    int start = 1;

    int end = 30;

    for (int i = start; i <= end; i += STEP\_SIZE) {

        printf("%d ", i);

    }

    printf("\n");

    return 0;

}

8

#include<stdio.h>

int main(){

    const int rows=3;

    const int columns=10;

    for(int i=0;i<rows;i++){

        for(int j=0;j<columns;j++){

            printf("\*");

        }

        printf("\n");

    }

}

-------------------------------------------------storage classes--------------------------------------------------

1

#include<stdio.h>

void cal\_sum(){

    static int sum=0;

    for(int i=0;i<=10;i++){

        sum+=i;

    }

    printf("The sum for the iteration is %d\n",sum);

}

int main(){

    for(int i=0;i<4;i++){

        cal\_sum();

    }

}

2

#include<stdio.h>

int loop\_run(){

    static int count=0;

    for(int i=0;i<5;i++){

        count+=1;

    }

    printf("the total count is %d\n",count);

}

int main(){

    loop\_run();

    loop\_run();

}

3

#include<stdio.h>

void inner\_loop(){

    static int count=0;

    for(int i=0;i<4;i++){

        for(int j=0;j<4;j++){

            count+=1;

        }

    }

    printf("total inner executions=%d\n",count);

}

int main(){

    inner\_loop();

    inner\_loop();

}

4

#include<stdio.h>

void count\_times(){

    static int count=0;

    for(int i=1;i<10;i++){

        if(i%2!=0){

            count+=1;

            break;

        }

        }

    printf("the value of count = %d",count);

    printf("\n");

}

int main(){

    count\_times();

    count\_times();

}

5

#include<stdio.h>

void loop\_reentry(){

    static int count=0;

    for(int i=1;i<10;i++){

        if(i%2==0){

            count+=1;

            break;

        }

    }

    printf("the reentry count is %d\n",count);

}

int main(){

    loop\_reentry();

    loop\_reentry();

    loop\_reentry();

    loop\_reentry();

}

6

#include<stdio.h>

void step\_count() {

    static int total\_steps = 0;

    int normal\_steps = 0;

    int step\_size = 1;

    for (int i = 1; i < 10; i += step\_size) {

        normal\_steps++;

        step\_size++;

    }

    total\_steps += normal\_steps;

    printf("The total number of steps taken in this loop = %d\n", normal\_steps);

    printf("The total steps throughout the iteration = %d\n", total\_steps);

    printf("\n");

}

int main() {

    step\_count();

    step\_count();

    step\_count();

    step\_count();

    return 0;

}

--------------------------------------------------------scope of variables----------------------------------------  
1

#include<stdio.h>

int x=10;

void modify\_variables(){

    int x=5;

    printf("the value inside the function is:%d\n",x);

}

int main(){

  printf("before the function call=%d\n",x);

  modify\_variables();

  printf("after the function call=%d\n",x);

  return 0;

}

2

#include<stdio.h>

int x=10;

void addition(){

    int x=10;

    printf("adding 5 to x gives the value =%d\n",x+5);

}

void substraction(){

    int x=10;

    printf("substracting 5 from the value gives =%d\n",x-5);

}

void multiplying(){

    int x=10;

    printf("multiplying 5 to the value gives =%d\n",x\*5);

}

int main(){

    printf("the values before doing the operations=%d\n",x);

    addition();

    substraction();

    multiplying();

    printf("the values after doing the operations=%d\n",x);

}

3

#include<stdio.h>

void func1(){

    int x=10;

    printf("the value of x is %d\n",x);

}

int main(){

    for(int i=1;i<4;i++){

        printf("Calling the function %d time\n",i);

        func1();

    }

    return 0;

}

4

#include<stdio.h>

int x=10;

void sum(){

    int y=5;

    int sum=x+y;

    printf("the result is %d",sum);

}

int main(){

    sum();

    return 0;

}

5

#include<stdio.h>

int counter=2;

int increment\_counter(){

    counter+=1;

    return counter;

}

int main(){

    for(int i=1;i<4;i++){

        int result=increment\_counter();

        printf("the value of count in the %d call is %d\n",i,result);

    }

}

6

#include <stdio.h>

int x = 10;

void modify\_variables() {

    int x = 5;

    printf("The value of the local variable x inside the function: %d\n", x);

}

int main() {

    printf("The value of the global variable x before the function call: %d\n", x);

    modify\_variables();

    printf("The value of the global variable x after the function call: %d\n", x);

    return 0;

}

7

#include <stdio.h>

const int GLOBAL\_CONST = 100;

void print\_global\_constant() {

    printf("The value of the global constant inside print\_global\_constant: %d\n", GLOBAL\_CONST);

}

void modify\_global\_constant() {

    GLOBAL\_CONST = 200;

}

int main() {

    printf("The value of the global constant in main: %d\n", GLOBAL\_CONST);

    print\_global\_constant();

    return 0;

}

8

#include<stdio.h>

int value=100;

void print(){

    printf("%d\n",value);

}

void set\_new(int *new*){

    value=*new*;

    printf("%d\n",value);

}

void sum(int *input*){

    int sum=*input*+value;

    printf("%d\n",sum);

}

int main(){

    printf("the initial value is :\n");

    print();

    printf("the value after updation is:\n");

    set\_new(5);

    printf("the sum is:\n");

    sum(10);

    return 0;

}

9

#include<stdio.h>

int x=10;

void local\_declaration(){

    int y=5;

}

int main(){

    printf("the value of x is:%d",x);

    printf("the value of y is:%d",y);

    return 0;

}

10

#include<stdio.h>

int gobal\_total=0;

void calculate\_sum(int *arr*[],int *size*){

    int local\_total=0;

    for(int i=0;i<*size*;i++){

        local\_total+=*arr*[i];

    }

    gobal\_total+=local\_total;

    printf("local sum=%d\n",local\_total);

}

int main(){

    int arr1[]={10,20,30},size1=sizeof(arr1)/sizeof(arr1[0]);

    int arr2[]={1,2,3},size2=sizeof(arr1)/sizeof(arr1[0]);

    calculate\_sum(arr1,size1);

    calculate\_sum(arr2,size2);

    printf("the total sum is=%d",gobal\_total);

    return 0;

}