**六、指出以下程序或函数的功能**

1. #include<iostream.h>

void main()

{

int i,s=0;

for(i=2;i<=30;i+=2) s+=i\*i;

cout<<"s="<<s<<endl;

}

2. #include<iostream.h>

#include<stdlib.h>

#include<math.h>

void main()

{

int i=10,a;

while(i>0) {

a=rand()%100+10;

int j, k=int(sqrt(a)+1e-5); //sqrt(x)为求x的平方根函数

for(j=2; j<=k; j++)

if(a%j==0) break;

if(j>k) {cout<<a<<' '; i--;}

}

}

3. void trans(int x)

{

char a[10];

int i=0,rem;

do {

rem=x%16;

x=x/16;

if(rem<10) a[i]=48+rem; //’0’字符的ASCII码为48

else a[i]=55+rem; //’A’字符的ASCII码为65

i++;

}while(x!=0);

while(i>0) cout<<a[i];

cout<<endl;

}

4. #include<iostream.h>

double f1(int n) {

double sign=1,s=1;

for(int i=2;i<=n; i++) {

s+=sign/(i\*i);

sign\*=-1;

}

return s;

}

void main()

{

int a;

cin>>a;

cout<<f1(a)<<endl;

}

5. double f1(double a, double b, char op) {

switch(op) {

case ’+’: return a+b;

case ’-’: return a-b;

case ’\*’: return a\*b;

case ’/’: if(b==0) {

cout<<"divided by 0!"<<endl;

exit(1);

}

else return a/b;

default: cout<<"operator error!"<<endl;

exit(1);

}

}

6. #include<iostream.h>

#include<math.h>

void main()

{

int x,y;

cin>>x;

y=int(sqrt(x)); //sqrt(x)为求x的算术平方根

for(int i=1;i<=y;i++)

if(x%i==0) cout<<”x=”<<i<<’\*’<<x/i<<endl;

}

7. #include<iostream.h>

void main()

{

int i,p=1,s=0;

int N;

cout<<"输入一个正整数:";

cin>>N;

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

p\*=i;

s+=p;

}

cout<<s<<endl;

}

8. #include<iostream.h>

#include<stdlib.h>

#include<time.h>

const N=10;

int ff(int x, int y) {

int z;

cout<<x<<'+'<<y<<'=';

cin>>z;

if(x+y==z) return 1; else return 0;

}

void main()

{

int a,b,c=0;

srand(time(0)); //初始化随机数序列

for(int i=0;i<N;i++) {

a=rand()%20+1; //rand()函数产生0-32767之间的一个随机数

b=rand()%20+1;

c+=ff(a,b);

}

cout<<"得分:"<<c\*10<<endl;

}

9. int s1(int n)

{

int x;

if(n==1) x=1;

else x=s1(n-1)+n\*n;

return x;

}

10. void fun5(char\* a, const char\* b)

{

while(\*b) \*a++=\*b++;

\*a=0;

}

11. template<class T>

bool fun8(T a[], int n, T key)

{

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

if(a[i]==key) return true;

return false ;

}

12. void f2(double a[], int n)

{

int i; double sum=0;

for(i=0;i<n;i++) sum+=a[i];

sum/=n;

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

if(a[i]>=sum) cout<<a[i]<<’ ’;

cout<<endl;

}

13. void f4(char a[M][N])

{

int c1,c2,c3;

c1=c2=c3=0;

for(int i=0;i<M;i++)

if(strlen(a[i])<5) c1++;

else if(strlen(a[i])>=5 && strlen(a[i])<15) c2++;

else c3++;

cout<<c1<<’ ’<<c2<<’ ’<<c3<<endl;

}

14. void fun3(int a[][N], int m, int& row, int& col)

{

int x=a[0][0];

row=col=0;

for(int i=0;i<m;i++)

for(int j=0;j<N;j++)

if(a[i][j]>x) {

x=a[i][j]; row=i; col=j;

}

}

15. int fun6(int m, int n, int b=2)

{

if(m<b && n<b) return m\*n;

else if(m%b==0 && n%b==0) return b\*fun6(m/b,n/b,b);

else return fun6(m,n,++b);

}

16. char\* f8(char\* str1, const char\* str2)

{

int i=0,j=0;

while(str1[i]) i++;

while(str2[j]) str1[i++]=str2[j++] ;

str1[i]='\0';

return str1;

}

17. int f8(const char\* str1, const char\* str2)

{

int i=0;

while(str1[i] && str2[i])

if(str1[i]==str2[i]) i++;

else if(str1[i]>str2[i]) return 1;

else return -1;

if(str1[i]==str2[i]) return 0;

else if(str1[i]>str2[i]) return 1;

else return -1;

}

18. IntNode\* FindMax(IntNode \*f)

{

if(!f) return NULL;

IntNode \*p=f;

f=f->next;

while(f) {

if(f->data > p->data) p=f;

f=f->next;

}

return p;

}

假定IntNode的类型定义为：

struct IntNode {

int data; //结点值域

IntNode\* next; //结点指针域

};

19. int Count(IntNode \*f)

{

if(!f) return 0;

int c=0;

while(f) {

c++;

f=f->next;

}

return c;

}

假定IntNode的类型定义为：

struct IntNode {

int data; //结点值域

IntNode\* next; //结点指针域

};

20. void Output(IntNode \*f)

{

if(!f) return;

while(f) {

cout<<f->data<<’ ’;

f=f->next;

}

cout<<endl;

}

假定IntNode的类型定义为：

struct IntNode {

int data; //结点值域

IntNode\* next; //结点指针域

};

21. void Input(IntNode\*& f)

{

int n;

cout<<”从键盘给n输入一个整数:”;

do cin>>n; while(n<0);

if(n==0) {f=NULL; return;}

f=new IntNode;

IntNode\* p=f;

cout<<”从键盘输入”<<n<<”个整数:”;

while(n--) {

p=p->next=new IntNode;

cin>>p->data;

}

p->next =NULL;

p=f; f=f->next; delete p;

}

假定IntNode的类型定义为：

struct IntNode {

int data; //结点值域

IntNode\* next; //结点指针域

};

22. int f(const char \*s)

{

int i=0;

while(\*s++)i++;

return i;

};

23. char \*f(char \*s){

int n=strlen(s);

char\* r=new char[n+1];

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

if(s[i]>='a' && s[i]<='z') r[i]=s[i]-'a'+'A';

else r[i]=s[i];

r[n]=’\0’;

return r;

}