1. 什么是算法？试从日常生活中找3个例子，描述它们的算法

算法：算法就是解决一个问题的方法及具体步骤。

例子：1. 考四级：先积累单词，再刷题，去考场。

​ 2.包饺子：先准备馅和面，再擀饺子皮，包饺子。

​ 3.下面条：先煮开水，再将面条放入开水中。

第二题

2.什么叫结构化的算法？为什么要提倡结构化的算法？

（1）由顺序，选择，循环等基本结构组成的算法结构是结构化的算法。

（2）结构化的算法可读性高，不易出现错误，使算法的质量得到保证和提高，所以要提倡结构化的算法。

第三题

3.试述3种基本结构的特点，请另外设计两种基本结构(要符合基类结构的特点)。

（1）顺序结构：顺序结构是一种线性，有序的结构，它依次执行各种语句模块。

（2）选择结构：选择结构是根据条件成立与否选择程序执行的通路。

（3）循环结构：循环结构是重复执行一个或几个模块，直到满足某一条件位置。

基本结构要满足以下几点：

只有一个入口

只有一个出口

结构内的每一部分都有机会执行到

结构内不存在死循环

因此给出以下结构：

第四题

4.用传统流程图表示求解以下问题的算法。

(1)有两个瓶子A和B，分别盛放醋和酱油，要求将他们互换(即A瓶原来盛醋，现在盛酱油，B瓶则相反)。

思路：两个瓶子，要实现交换内容，必须再用一个新瓶子来做媒介。

所以流程图如下：

代码如下：

#include<stdio.h>

#include<string.h>

int main()

{

char a[10]="cu";//醋

char b[10]="jiangyou";//酱油

char c[10]={0};

strcpy(c,a);

strcpy(a,b);

strcpy(b,c);

printf("%s %s\n",a,b);

return 0;

}

(2)依次将10个数输入，要求输出其中最大的数。

思路：将第一个数赋给m，循环9次，依次与后面的数比较，如果小于后面的值，就将m的值替换为此值，循环结束后，最大值就是m，也就是10个数中最大的数。

所以流程图如下：

代码如下：

#include<stdio.h>

int main()

{

int arr[10]={0};

for(int i=0;i<10;i++)

{

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

}

int m=arr[0];

for(int i=1;i<10;i++)

{

if(m<arr[i])

{

m=arr[i];

}

}

printf()

return 0;

}

（3）有3个数a,b,c, 要求按大小顺序把他们输出。

思路：输入的三个数，a，b，c存起来，先a和b进行比较，如果a大于b，那么a和b的值互换。a和c进行比较，如果a大于c，那么a和c的值互换。b和c比较如果b大于c的值，b和c的值互换。结果就是a，b，c从小到大排序。

所以流程图如下：

代码如下：

#include<stdio.h>

int main()

{

int a,b,c,tmp;

scanf("%d%d%d",&a,&b,&c);

if(a>b)

{

tmp=a;

a=b;

b=tmp;

}

if(a>c)

{

tmp=a;

a=c;

c=tmp;

}

if(b>c)

{

tmp=b;

b=c;

c=tmp;

}

printf("a=%d b=%d c=%d\n",a,b,c);

return 0;

}

(4)求1 + 2 + 3 + … + 100。

思路：这是一个累加，从1到100，只需要一个循环即可。

流程图如下：

代码如下：

#include<stdio.h>

int main()

{

int sum=0;

for(int i=1;i<=100;i++)

{

sum+=i;

}

printf("%d\n",sum);

retrun 0;

}

(5)判断一个数n能否同时被3和5整除。

思路：输入一个n，先判断能否被3整除，不能就表示不满足条件，如果能就再判断是否可以被5整除，不能就不满足条件，如果能表明此数能同时被3和5整除。

流程图如下：

代码如下：

int main()

{

int n=0;

scanf("%d",&n);

if(n%3==0)

{

if(n%5==0)

printf("满足条件\n");

else

printf("不满足条件\n");

}

else

printf("不满足条件\n");

retrun 0;

}

15

(6)将100~200之间的素数输出

思路：素数是除了本身和1，不能被其他数整除。所以只需用100~200的数去除以小于它的数（不包括1），再判断能否整除即可。

流程图如下：

#include<stdio.h>

int Panduansushu(int n)

{

for(int i=2;i<b;i++)

{

if(n%i==0)

return (-1);

}

return 1;

}

int main()

{

for(int i=100;i<=200;i++)

{

int ret=Panduansushu(i);

if(i!=(-1))

printf("%d\n",i);

}

return 0;

}

(7)求两个数m和n的最大公约数

思路：求俩个数的最大公约数可以用辗转相除法。

流程图如下：

代码如下：

int main()

{

int n,m;

scanf("%d%d",&n,&m);

if(m>n)

{

int tmp=m;

m=n;

n=tmp;

}

for(int i=n;i>0;i--)

{

if((m%i==0)&&(n%i==0))

printf("%d\n",i);

}

}

（8）求方程的解：

a x 2 + b x + c = 0 ; ax^2+bx+c=0;

ax

2

+bx+c=0;

思路：这是一道初中数学题。

流程图如下：

代码如下：

#include<stdio.h>

#include<math.h>

int main()

{

float a, b; float x1, x2; float c, t;

while ((scanf("%f %f %f\n", &a, &b, &c)) != EOF)//多组输入系数a，b，c

{

if (a == 0)

printf("Not quadratic equation\n");

else

{

t = (b\*b) - 4 \* a\*c;//t就是用来判断根的情况（derta~）

if (t>0)//两个根

{

//求根公式sqrt（t）是开平方

x1 = (-b - sqrt(t)) / (2 \* a);

x2 = (-b + sqrt(t)) / (2 \* a);

printf("x1=%.2f;x2=%.2f\n", x1, x2);

}

else if (t<0)//这是复数根

{

x1 = -b / (2 \* a);

x2 = sqrt(-t) / (2 \* a);

printf("x1=%.2f-%.2fi;x2=%.2f+%.2fi\n", x1, x2, x1, x2);

}

else //唯一根

{

x1 = (-b) / (2 \* a);

printf("x1=x2=%.2f\n", x1);

}

}

}

return 0;

}

第七题

7. 什么叫结构化程序设计？它的主要内容是什么？

结构化程序设计\*\*（structured programming，简称SP）是进行以模块功能和处理过程设计为主的详细设计的基本原则。其概念最早由E.W.Dijikstra在1965年提出的。结构化程序设计思想确实使程序执行效率提高 ，是软件发展的一个重要的里程碑，它的主要观点是采用自顶向下、逐步求精的程序设计方法；各个模块通过“顺序、选择、循环”的控制结构进行连接，并且只有一个入口、一个出口 。