**Câu 1 ( 3đ):**

a. In ra màn hình các thông tin sau: Tên của bạn , ngày sinh, mã số sinh viên , lớp.

Source code:

#include<stdio.h>

int main()

{

printf("toi ten la: Vu Duc Viet");

printf("\nNgay sinh: 11/05/2003");

printf("\nmssv: 21105481");

printf("\nDHIOT17B");

return 0;

}

Text

Description automatically generated

b. In ra màn hình bảng cửu chương nhân (dùng for) ,chia (dùng do while) theo n.

Source code:

#include<stdio.h>

int main()

{

int n;

printf("Nhap bang cuu chuong: ");

scanf("%d",&n );

printf("\nBang cuu chuong nhan:");

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

{

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

}

printf("\n =======================");

printf("\nBang cuu chuong chia:");

int i = 1;

while( i <= 10)

{

printf("\n%d / %d = %d",n\*i,n,i);

i++;

}

return 0;

}

Text

Description automatically generated

c. Viết 1 chương trình in ra cây thông theo chiều cao. Thân cây thông cao gấp 5 lần Chân cây Bamnthông.

#include<stdio.h>

#include<math.h>

int main(int argc, char \*argv[])

{

int h;

printf("nhap h: ");

scanf("%d",&h);

for( int i=1; i<=h; i++){

for (int j=1; j<h\*2; j++){

if (abs(h- j)<=( i-1))

printf(" \* ");

else printf(" ");

}

printf("\n");

}

for (int i=1; i<=h/5;i++){

for (int j=1; j<=h-1;j++){

printf(" ");

}

for (int k=1; k<=1 ;k++){

printf(" \* ");

}

printf("\n");

}

return 0;

}

A picture containing graphical user interface

Description automatically generated

**Câu 2  ( 3đ) :**

 Viết chương trình nhập vào tổng số tiền, biết rằng có bốn loại mệnh giá tiền là 500.000VNĐ, 200.000VNĐ,100.000VNĐ, 50.000VNĐ. Cho biết tổng số tiền trên có ít nhất là bao nhiêu tờ từ các mệnh giá đã biết.Và xuất ra số dư còn lại (nếu có)

 Ví dụ số tiền là: 2.850.000 có 5 tờ 500.000, 1 tờ 200.000, 1 từ 100.000, 1 từ 50.000. Phương án trên có tổng số tờ là 8 và đây là phương án có số tờ tiền ít nhất.

#include<stdio.h>

#include<conio.h>

int main()

{

int t500, t200,t100,t50,tt;

printf("nhap tong tien:");

scanf("%d",&tt);

t500= tt/500000;

tt-=t500\*500000;

t200= tt/200000;

tt-=t200\*200000;

t100= tt/100000;

tt-=t100\*100000;

t50= tt/50000;

tt-=t50\*50000;

printf("so to 500000 la: %d",t500);

printf("\nso to 200000 la: %d",t200);

printf("\nso to 100000 la: %d",t100);

printf("\nso to 50000 la: %d",t50);

printf("\nso tien thua: %d",tt);

printf("\ntong so to la: %d",t500+t200+t100+t50);

return 0;

}

Text

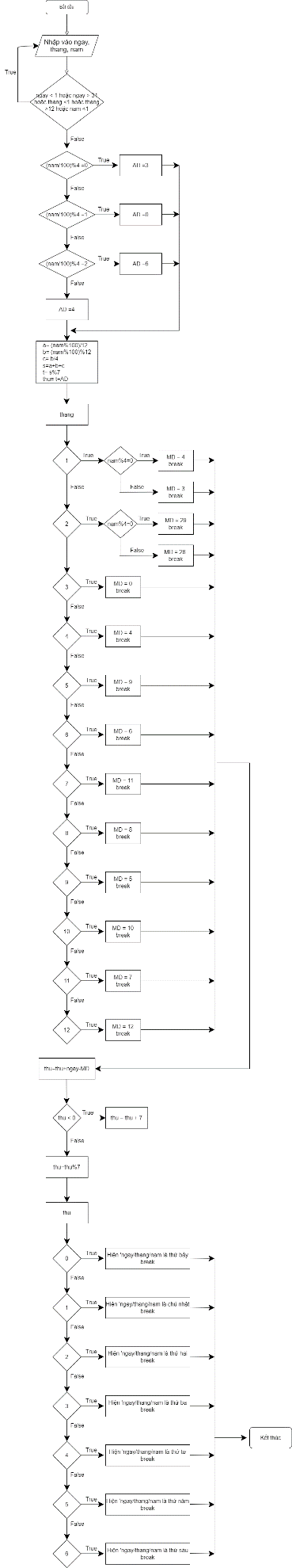
Description automatically generated

**Câu 3 ( 4đ) :**

a. Vẽ lưu đồ giải thuật (1d)

b. Nhập ngày/tháng/năm, cho biết ngày đó là ngày thứ mấy trong tuần. (Gợi ý: thuật toán Doomsday)

Lưu đồ giải thuật:



source code:

#include<stdio.h>

#include<math.h>

int main(int argc, char \*argv[])

{

int ngay, thang, nam;

int AD,MD;

do{

printf("\nNhap ngay: ");

scanf("%d",&ngay);

printf("\nNhap thang: ");

scanf("%d",&thang);

printf("\nNhap nam: ");

scanf("%d",&nam);

}

while ((ngay<1)||(ngay>31)||(thang<1)||(thang>12)||(nam<0));

if ((nam/100)%4 == 0)

AD=3;

else if ((nam/100)%4 == 1)

AD=0;

else if ((nam/100)%4 == 2)

AD=6;

else AD=4;

int a= (nam%100)/12;

int b= (nam%100)%12;

int c= b/4;

int s=a+b+c;

int t= s%7;

int thu= t+AD;

switch (thang)

{

case 1:

if (nam%4==0) MD=4;

else MD=3;break;

case 2:

if (nam%4==0) MD=29;

else MD=28;break;

case 3: MD=0;break;

case 4: MD=4;break;

case 5: MD=9;break;

case 6: MD=6;break;

case 7: MD=11;break;

case 8: MD=8;break;

case 9: MD=5;break;

case 10: MD=10;break;

case 11: MD=7;break;

case 12: MD=12;

break;

}

thu=thu+ngay-MD;

if(thu<0) thu+=7;

thu=thu%7;

switch (thu)

{

case 0: printf ("\n%.2d/%.2d/%d la thu bay",ngay,thang,nam);

break;

case 1: printf ("\n%.2d/%.2d/%d la chu nhat",ngay,thang,nam);

break;

case 2: printf ("\n%.2d/%.2d/%d la thu hai",ngay,thang,nam);

break;

case 3: printf ("\n%.2d/%.2d/%d la thu ba",ngay,thang,nam);

break;

case 4: printf ("\n%.2d/%.2d/%d la thu tu",ngay,thang,nam);

break;

case 5: printf ("\n%.2d/%.2d/%d la thu nam",

ngay,thang,nam);

break;

case 6: printf ("\n%.2d/%.2d/%d la thu sau",ngay,thang,nam);

break;

}

return 0;

}

Text

Description automatically generated with low confidence