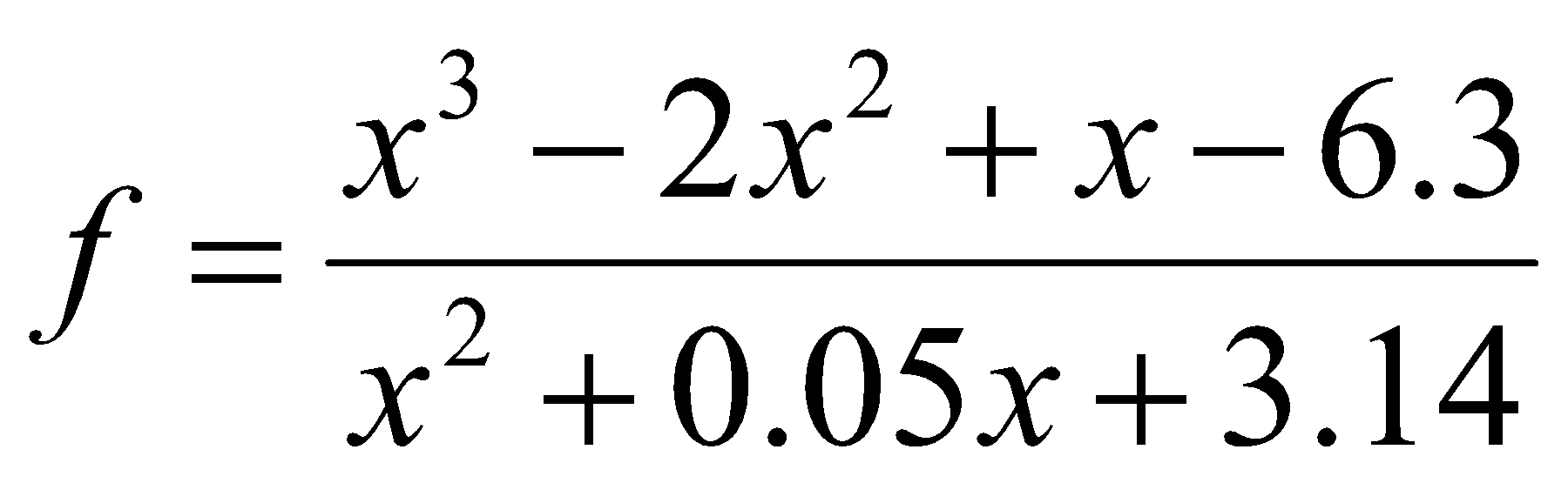
**Students are instructed to write algorithm for the problems in the observation note. In the observation note for each problem record minimum 5 test cases.**

**Also maintain error log**

1. Write a C program to compute the following



**CODE :**

#include<stdio.h>

#include<math.h>

void main()

{

int x;

float f;

printf("Enter Value for x:");

scanf("%d",&x);

f = (pow(x,3)-(2\*(pow(x,2))+x-6.3))/(pow(x,2)+0.05\*x+3.14);

printf("ANSWER:%f",f);

}

1. Write a C program to print the roots of a quadratic equation.

CODE :

#include<stdio.h>

#include<math.h>

void main()

{

int a,b,c;

float x1,x2;

printf("Enter value of a:");

scanf("%d",&a);

printf("Enter value of b:");

scanf("%d",&b);

printf("Enter value of c:");

scanf("%d",&c);

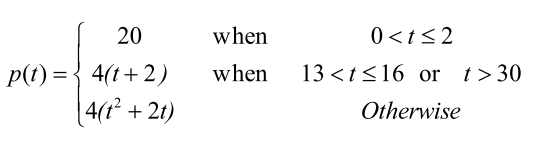
x1 = ((-b)+sqrt(pow(b,2)-(4\*a\*c)))/(2\*a);

x2 = ((-b)-sqrt(pow(b,2)-(4\*a\*c)))/(2\*a);

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

}

1. Write the complete C program that asks the user to enter the value for t from the keyboard and then it computes and prints the value of p which is expressed as a function of t by



**CODE :**

#include<stdio.h>

#include<math.h>

void main()

{

int t,f1,f2;

printf("Enter Value of t:");

scanf("%d",&t);

if (t>0 && t<=12){

printf("20");

}

else if((t>=13 && t<=16) || t>30){

f1= 4\*(t+2);

printf("Answer=%d",f1);

}

else{

f2=4\*(pow(t,2)+2\*t);

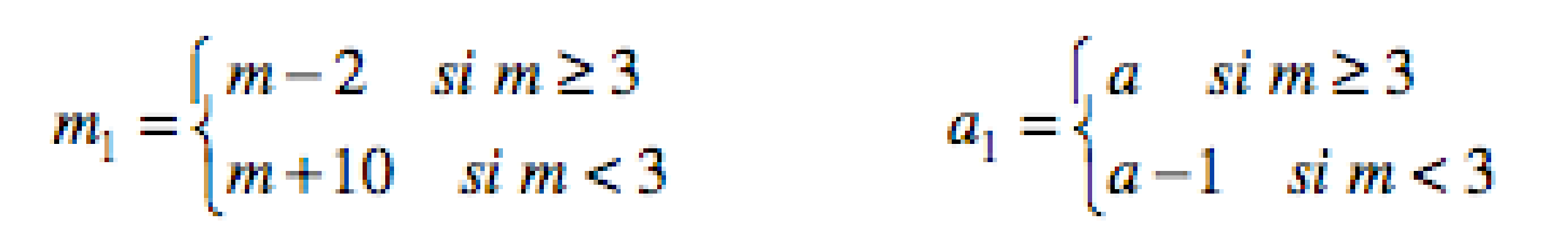
printf("Answer=%d",f2);

}

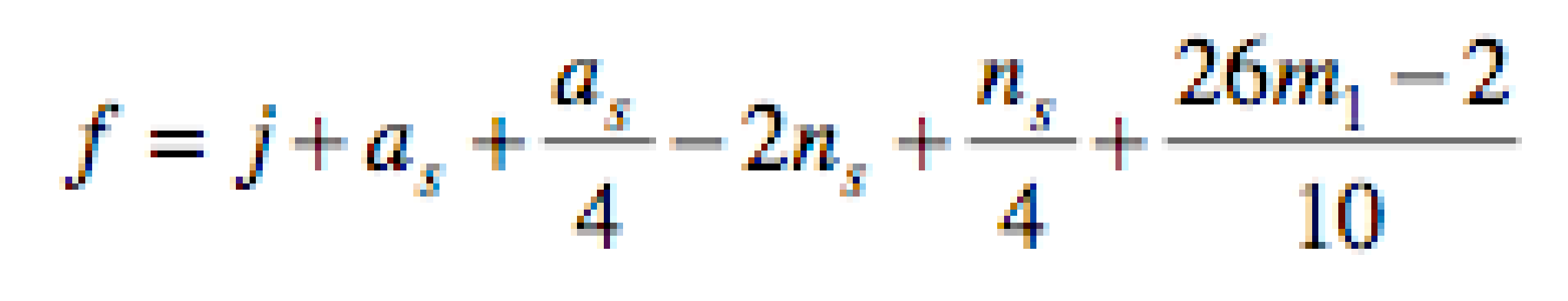
return 0;

}

1. Write a program that outputs the day of the week given a date expressed as *j* (day) *m* (month) *a* (year). You will use the following formula:



and with *ns* being the two first digits of *a1* and *as* the two last digits of *a1*



The day of the week will then be given by the modulo of *f* and 7 (0 is Sunday, 1 Monday etc). Let the date be DD/MM/CCYY (european format), where DD is the day of the month, MM is the month, CC the century-digits and YY the year within the century. The for the date 23/06/1994. Starting with the century CC-digits, calculate CC/4 - 2\*CC-1 and remember the result. With all divisions in this exercise, discard any remainder and just keep the whole part. So, in our example, this is 19/4=4 minus 2\*19=38 minus 1, giving minus 35.

Now, using the year YY, calculate 5\*YY/4. In this example that's 5\*94 = 470/4 = 117, discarding the remainder. Adding this to our existing result gives 117-35 = 82.

Using the month MM, calculate 26\*(MM+1)/10. In our example this is 26\*7 = 182 / 10 = 18, again discarding the remainder. Add this to our running total giving 82+18 = 100.

Finally just add the day DD. Here 100 + 23 = 123.

Now divide the result by 7, just **keeping the remainder**; here 123(mod 7) = 4. Counting Sunday as zero, Monday = 1 etc, we get 4 = Thursday. *Easy, when you know how :-)*

**CODE:**

#include<stdio.h>

void main(){

int j,m,a,as,ml,f,al,ns,result;

printf("Enter a day /month/year:");

scanf("%d/%d/%d",&j,&m,&a);

if(m >= 3){

ml = m-2;

al = a;

}

else{

ml=m+10;

al=a-1;

}

ns=al/100;

as = al%100;

f = j+as+(as/4)-(2\*ns)+(ns/4)+(((26\*ml)-2)/10);

result = f%7;

switch(result){

case 0:

printf("Day of week is Sunday\n");

break;

case 1:

printf("Day of week is Monday\n");

break;

case 2:

printf("Day of week is Tuesday\n");

break;

case 3:

printf("Day of week is Wednesday\n");

break;

case 4:

printf("Day of week is Thursday\n");

break;

case 5:

printf("Day of week is Friday\n");

break;

case 6:

printf("Day of week is Saturday\n");

break;

}

printf("Result=%d",result);

}

1. Write a program to find the biggest among two numbers without using any control structures.

**CODE:**

#include<stdio.h>

void main(){

int a , b , c;

printf("Enter a value of a & b:");

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

c = (a+b+abs(a-b))/2;

printf("Highest number is:%d",c);

}

1. Write a C program to print \* in the format given below.

|  |  |  |  |
| --- | --- | --- | --- |
| Right Triangle | Pyramid | Diamond | Right Arrow |
| **http://www.arraypeers.com/files/Editor_images/Capture3.PNG** | http://www.arraypeers.com/files/Editor_images/pattern_figure.PNG | **http://www.arraypeers.com/files/Editor_images/Capture2.PNG** | **http://www.arraypeers.com/files/Editor_images/Capture8.PNG** |

#include<stdio.h>

void main(){

int a, i ,j,space;

printf("Right Triangle:");

for(i = 0 ; i <= 5 ; i++){

printf("\n");

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

printf("\*");

}

}

printf("\nPyramid Triangle:\n");

for(i=1;i<=5;i++){

for(space=1;space<=5-i;space++){

printf(" ");

}

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

printf("\*");

}

printf("\n");

}

printf("Diamond Triangle:\n");

for (i = 1; i <= 5; i++) {

for (space = 1; space <= 5 - i; space++)

printf(" ");

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

printf("\*");

printf("\n");

}

for (i = 5 - 1; i >= 1; i--) {

for (space = 1; space <= 5 - i; space++)

printf(" ");

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

printf("\*");

printf("\n");

}

printf("Right Arrow Triangle:")

for (i = 0; i < 5; i++) {

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

printf(" ");

for (j = 0; j < 5 - i; j++)

printf("\*");

printf("\n");

}

for (i = 1; i < 5; i++) {

for (j = 0; j < 5 - i - 1; j++)

printf(" ");

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

printf("\*");

printf("\n");

}

}

1. Write a program for a matchstick game being played between the computer and a user. Your program should ensure that the computer always wins. Rules for the game are as follows:

− There are 21 matchsticks.

− The computer asks the player to pick 1, 2, 3, or 4 matchsticks.

− After the person picks, the computer does its picking.

− Whoever is forced to pick up the last matchstick loses the game.

#include <stdio.h>

int main() {

int matchsticks = 21, user, computer;

while (matchsticks > 1) {

printf("Matchsticks left: %d\n", matchsticks);

printf("Pick 1, 2, 3, or 4 matchsticks: ");

scanf("%d", &user);

if (user < 1 || user > 4) {

printf("Invalid choice. Try again.\n");

continue;

}

computer = 5 - user;

printf("Computer picks: %d\n", computer);

matchsticks -= (user + computer);

}

printf("Only 1 matchstick left. You lose!\n");

return 0;

}

1. A number is said to be perfect if it is equal to the sum of all numbers which are its factors (excluding itself). So, for example, 6 is perfect, because it is the sum of its factors 1,2,3. Write a program which determines if a number is perfect. It should also print its factors.

#include <stdio.h>

int main() {

int n, i, sum = 0;

printf("Enter a number: ");

scanf("%d", &n);

printf("Factors: ");

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

if (n % i == 0) {

printf("%d ", i);

sum += i;

}

}

printf("\n");

if (sum == n)

printf("%d is a perfect number\n", n);

else

printf("%d is not a perfect number\n", n);

return 0;

}

1. Write a program that takes as input a natural number x and prints the smallest palindrome larger than x. A palindrome is a word, number, phrase, or other sequence of characters which reads the same backward as forward, such as madam, racecar. There are also numeric palindromes, including date/time stamps using short digits 11/11/11 11:11 and long digits 02/02/2020.

Input: n = 121  
 Output: Next smallest palindrome of 121 is 131

#include <stdio.h>

int main() {

int n, next, temp, reversed;

printf("Enter a number: ");

scanf("%d", &n);

next = n + 1;

while (1) {

temp = next;

reversed = 0;

while (temp > 0) {

reversed = reversed \* 10 + temp % 10;

temp /= 10;

}

if (reversed == next) {

printf("Next smallest palindrome of %d is %d\n", n, next);

break;

}

next++;

}

return 0;

}

1. According to a study, the approximate level of intelligence of a person can be calculated using the following formula: i = 2 + ( y + 0.5 x )

Write a program, which will produce a table of values of i, y and x, where y varies from 1 to 6, and, for each value of y, x varies from 5.5 to 12.5 in steps of 0.5.

#include <stdio.h>

int main() {

int y;

float x, i;

printf(" y\t x\t i\n");

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

for (y = 1; y <= 6; y++) {

for (x = 5.5; x <= 12.5; x += 0.5) {

i = 2 + (y + 0.5 \* x);

printf(" %d\t%.1f\t%.2f\n", y, x, i);

}

}

return 0;

}

**Deadline: 19.08.2024**