Tentamen - Programmering

DVA117

----------FACIT----------

Akademin för innovation, design och teknik

Onsdag 2020-08-12

An English translation of the entire exam follows after the questions in Swedish

Skrivtid: 08.10 – 13.30
Hjälpmedel: Valfritt icke-elektroniskt material
Lärare: Caroline Uppsäll
(kan nås på telefon om du frågar tentavakten)

Preliminära betygsgränser

Betyg 3:	22p
Betyg 4:	32p
Betyg 5:	38p
Max:	**43p**

Allmänt

- All kod skall skrivas i C.
- Skriv tydligt vilken uppgift/deluppgift ditt svar anser.
- Skriv endast på bladets **ena** sida.
- Referera inte mellan olika uppgifter.
- Om du är osäker på vad som avses i någon fråga, skriv då vad du gör för antagande.
- Oläsliga/oförståeliga/ostrukturerade svar rättas inte.
- Kommentera din kod.
- Det är inte tillåtet att använda goto-satser och globala variabler
- Tips: Läs igenom hela tentamen innan du börjar skriva för att veta hur du ska disponera din tid.
- Potentiellt intjänade bonuspoäng kan endast användas på kursinstansens ordinarie tentamen, alltså på denna tentamen för studenter som läser kursen HT19 Period 1.
- Förklaring som till viss del är rätt och till viss del är fel ger inga poäng. Otillräcklig förklaring/beskrivning ger inga poäng.

Lycka till!
/Caroline

----------FACIT----------

Notera: alla lösningar bedöms individuellt och förändringar från nedanstående poängsättning (inom uppgiften) kan förekomma beroende på hur resterande svar ser ut.

Uppgift 1 [5p]

Hur ser utskriften ut när följande kod kört färdigt om inputen är ”Salmon sushie dish”?

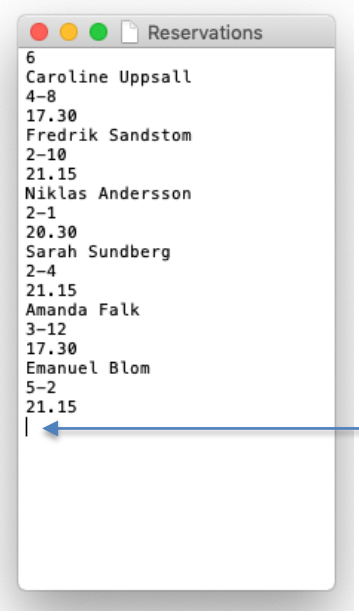
```c
#include <stdio.h>
#include <string.h>
#define STRMAX 30

int main(void)
{
    char str[STRMAX], strNew[STRMAX*2];
    int i;
    printf("Enter a string: ");
    fgets(str, STRMAX, stdin); //Läs in strängen
    str[strlen(str)-1] = '\0';

    for(i = 0; str[i] != '\0'; i++) //gå igenom hela, till strängslut
    {
        strNew[i] = str[i];//kopiera över tecknet
        if(strNew[i] == 's' || strNew[i] == 'S')//om s eller S
            strNew[++i] = 's'; /*lägg till ett s efter (hoppar över nästa
                                 bokstav i str */
    }
    strNew[i] = '\0'; //lägg in strängslut i slutet

    puts(strNew); //skriv ut

    return 0;
}
```

Svar:
Sslmon ssssie diss

Grovt bedömningsförslag
Ssalmon → 1p
Ssalmon ssusshie → 2p
Ssalmon ssusshie dissh -> 4p

Uppgift 2 [14p]

Antag att du har följande fil (som heter Reservations) som innehåller reservationerna gjorda för en dag på en restaurang. Du ska i kommande uppgifter jobba med denna information på olika sätt.



Detta är textmarkören för att visa att det finns ett enterlsag på slutet av filen. I övrigt finns inte gömda tecken i filen. Det kan dock finnas tecken som syns men som ändå är lätta att missa.

På första raden i filen står hur många reservationer som finns nedskrivna i filen.
Varje reservation beskrivs sedan med följande information (på vardera rad):

Namn på personen som gjort reservationen
Antal personer i sällskapet-bordsnummer
Tid då sällskapet bokat bord

Den första reservationen i filen är alltså gjord av Caroline Uppsall, det är fyra personer i sällskapet och de har fått bordsnummer 8. Tiden för reservationen är 17.30.

a) Skapa en strukt som kan hålla informationen om en reservation som finns i filen ovan. Strukten ska heta ”struct Reservation”. [2p]

Lösningsförslag (2p)

```
struct Reservation
{
    char name[NAMEMAX];
    int nrOfPeople
    int tableNr;
    float time
};
```

b) Öppna filen (Reservations) och läs in all information som finns i filen (om de olika reservationerna) och spara den i en dynamiskt allokerad array. Pekaren till den dynamiska arrayen ska vara av typen du skapade i a) och ska heta todaysReservations. [7p]

Lösningsförslag (7p)

```
struct Reservation *todaysReservations;
int nrOfReservations;
FILE *fp = fopen("/Users/cul01/Desktop/Reservations", "r"); (1p)
if(fp != NULL)    (0.5p)
{
    fscanf(fp, "%d\n", &nrOfReservations); (0.5p)
    todaysReservations = (struct Reservation*)calloc(nrOfReservations,
                                                     sizeof(struct Reservation)); (1p)
    if(todaysReservations != NULL) (0.5p)
    {
        for(int i = 0; i < nrOfReservations; i++) (1p)
        {
            fgets(todaysReservations[i].name, NAMEMAX, fp); (0.5p)
            //todaysReservations[i].name[strlen(todaysReservations[i].name)-1] = '\0';
            fscanf(fp, "%d-%d\n", &todaysReservations[i].nrOfPeople,
                                  &todaysReservations[i].tableNr); (1p)
            fscanf(fp, "%f\n", &todaysReservations[i].time); (0.5p)
        }
    }
    fclose(fp); (0.5p)
```

c) Skriv en rad kod som ändrar bordsnumret på reservationen på index 2 (Niklas Andersson) till 3. Du måste använda piloperatorn (->). Du ska inte göra ändringen i filen utan i den inlästa informationen som ligger sparad i programmet från deluppgift b. Har du inte gjort deluppgift b går det ändå bra att göra denna deluppgift. [1p]

Lösningsförslag (1p)

```
(todaysReservations+2)->tableNr = 3;
```

d) Skapa en funktion som skriver hela listan med reservationer binärt till en fil. Du ska använda följande funktionshuvud: [4p]

```
void saveToBinaryFile(struct Reservation *allReservations, int nrOfReservations, char *filename)
```

Till parametern allReservations ska hela arrayen av reservationer skickas. Till parametern nrOfReservations ska antalet reservationer i arrayen skickas och till fileName ska det filnamn användaren vill att den skapade filen ska heta skickas.

Har du inte gjort deluppgift b som läser in informationen går det ändå bra att göra denna deluppgift.

Lösningsförslag (4p)

```
void saveToBinaryFile(struct Reservation *allReservations, int nrOfReservations,
                                                                    char *filename)
{
    FILE *fp = fopen(filename, "wb"); (1p)
    if(fp != NULL) (0.5p)
    {
        fwrite(&nrOfReservations, 1, sizeof(int), fp); (1p)
        fwrite(allReservations, nrOfReservations, sizeof(struct Reservation),  fp); (1p)
    }
    fclose(fp); (0.5p)
}
```

Uppgift 3 [5p]

Hur ser utskriften ut när följande program kört färdigt?

```c
#include <stdio.h>

int main(void)
{
    int a = 1, b = 2, i = 3;
    int arr[] = {5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60};
    int *p = &b, *pArr = arr;
    while (i < 10) {
        a = *pArr;
        pArr++;
        if (i < 4 || i == 8)
        {
            i += 3;
            p = &b;
        }
        else
        {
            i++;
            p = &a;
            (*p)++;
        }
        *p = *p + 1;
    }
    printf("a = %d\nb = %d\ni = %d\n", a, b, i);
    return 0;
}
```

Svar:
a = 20 (2.5p)
b = 4 (1.5p)
i = 11 (1p)

Uppgift 4 [5p]

Vilka av nedanstående påståenden är inte sanna? Du får ge maximalt 10 svar, om du ger fler svar än så kommer endast de första 10 svaren att bedömmas.

1) I en variabel deklarerad med int kan man spara både negativa och positiva heltal
2) * används för att få reda på adressen där en variabel ligger i minnet
3) Dynamiskt allokerat minne finns tills det frigörs med funktionen free
4) Funktionen strlen ger storleken på en sträng fram till (men inte inklusive) strängslutstecknet
5) En pekarvariabel tar inte upp något minne
6) En funktion kan anropas flera gånger i ett program
7) malloc eller calloc används för att allokera minne på heapen
8) Char-typen kan endast användas för att skapa strängar
9) Om man vill jämföra två strängar måste strcmp användas
10) fgets kräver att stdlib biblioteket är inkluderat
11) En pekarvariabel innehåller en adress
12) Modulu (%) används för att dividera två heltal
13) En funktion måste alltid returnera någonting
14) do-while loopen är fördelaktig om satserna i loopkroppen ska köras minst en gång.
15) En loopkropp körs om loopvillkoret evaluerar till sant
16) För att jämföra två värden används =
17) Om arr är en deklarerad array är *(arr+i) samma sak som arr[i]
18) En array måste vara av char-typ
19) En strukt allokeras alltid på heapen

Uppgift 5 [2p]

Vad evaluerar följande uttryck till (sant/1 eller falskt/0) med följande variabeldeklarationer:

```
int a = 9, b = 4, x = 3, y = 13;
```

a)
```
(a % 2 == 0 && a/b <= x) || (y - a < b)

(a % 2 == 0 && a/b <= x) || (y - a < b)
(9 % 2 == 0 && 9/4 <= 3) || (13 - 9 < 4)
(  1   == 0 &&  2  <= 3) || (   4   < 4)
(    Falskt &&  Sant   ) || (   Falskt)
       Falskt              || Falskt
                         Falskt
```

b)
```
a/2 > b && !((y - b > a) && a % x == 0)

a/2 > b && !((y - b > a) && a % x == 0)
9/2 > 4 && !((13 - 4 > 9) && 9 % 3 == 0)
 4  > 4 && !((   9   > 9) &&   0   == 0)
Falskt  && !((   Falskt ) &&   Sant    )
Falskt  && !(          Falskt          )
Falskt  && Sant
        Falskt
```

heltalsdivision

Svar:

a) Falskt (0)

b) Falskt (0)

Uppgift 6 [4p]

För följande funktionsanrop, ange funktionshuvudet (dvs det som används när man definierar och deklarerar funktionen).

Antag följande egendefinierade typ

```
struct Car{
    char brand[10];
    int productionYear;
    float distance;
};
```

Utgå från att följande deklarationer finns i programmet. Du kan anta att alla variabler som används är satta till tillåtna värden.

```
struct Car myCar, *anotherCar;
float x, *y;
int array[10], z;
char letter;
```

Exempel:

Funktionsanrop:	Ange deklarationen för
`x = func(6.3, letter);`	`func()` svar: `float func(float x, char c);`

Förklaring: Eftersom func() tar ett flyttal och ett tecken som argument måste parametrarna vara av typerna float och char (vad de heter är mindre viktigt).
Returvärdet tas emot i x som är deklarerad som en float, därför måste func() ha float som returtyp.

Uppgift:

Funktionsanrop:	Ange deklarationen för
1) `anotherCar = func1(10);`	`func1()`
2) `func2(&x, array);`	`func2()`
3) `letter = func3(*(array+1), myCar.distance);`	`func3()`
4) `anotherCar->productionYear = func4(myCar.brand, 'a');`	`func4()`

Lösningsförslag

```
1) struct Car *func1(int a);
2) void func2(float *r, int arr[]);
3) char func3(int z, float x);
4) int func4(char *str, char l);
```

0.5p per korrekt returtyp och 0.5p per korrekt parameterlista

Uppgift 7 [8p]

Skriv ett program som låter användaren mata in heltal som du ska använda för att beräkna ett resultat på följande vis (resultatet ska börja räknas från 0)

- Om talet är jämnt ska talet subtraheras från resultatet
- Om talet är udda ska talet adderas till resultatet
- Om talet är jämnt delbart med 10 ska ett tal mellan 1 och 50 slumpas fram, det slumpade talet ska adderas till resultatet.
- Om talet är negativt ska det göras om till ett positivt tal och sedan ska resterande regler följas (ex. -13 blir 13, -7 blir 7, -98 blir 98 etc).
- Om talet är större än 100 ska det ignoreras.
- Om talet är 0 ska inmatningen avslutas och resultatet skrivas ut på skärmen.

Exempelkörning

```
Enter a number (0 to exit): 5
Enter a number (0 to exit): 20
Enter a number (0 to exit): 24
Enter a number (0 to exit): 104
Enter a number (0 to exit): -19
Enter a number (0 to exit): 10
Enter a number (0 to exit): 0

The result is 63
```

Beräkningen gjordes såhär:
0+5 → 5
5+23 → 28 (23 slumpas då 20 är jämnt delbart med 10)
28-24 →4
104 ignoreras
4+19 → 23 (-19 omvandlas till 19)
23 + 40 → 63 (40 slumpas då 10 är jämnt delbart med 10)

Lösningsförslag:

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(void)
{
    int number, result = 0; //(0.5p)
    srand(time(0));
    do{
        printf("Enter a number (0 to exit): ");
        scanf("%d", &number); //(0.5p)
        if(number < 0) //(0.5p)
            number *= -1; //(0.5p)
        if(number != 0 && number <= 100) //(1p)
        {
            if(number%10 == 0) //(0.5p)
                result += rand()%50+1; //(1p)
            else if(number%2 == 0) //(0.5p)
                result -= number; //(0.5p)
            else if(number%2 == 1) //(0.5p)
                result += number; //(0.5p)
        }
    }while(number != 0); //(1p)

    printf("The result is %d\n", result); //(0.5p)
    return 0;
}
```

Exam - Programming

DVA117

----------Solutions----------

School of Innovation, design and technology

Wednesday 2020-08-12

Writing time: 08.10 – 13.30
Aids: Any non-electronic material
Examiner: Caroline Uppsäll
(Can be reached by telephone if you ask the exam guard)

Preliminary grading limits
Grade 3: 22p
Grade 4: 32p
Grade 5: 38p
Max: 43p

Generally

- All code should be written in C.
- Write clearly what task/sup-task your answers consider.
- Do only use **one** side of the paper.
- Do not refer between questions.
- If you are unsure of a meaning of a question, write down your assumption.
- Unreadable/incomprehensible answers will not be marked.
- Comment your code.
- It is not allowed to use goto-statements or global variables
- Hint: To know how to allocate your time, read through the entire exam before you start writing.
- Explanations that are to some extent correct and to som extent wrong does not give any points. Insufficient explanations/descriptions gives no points.

----------Solutions----------

Good luck!
/Caroline

---------- Solutions ----------

Note: all solutions are assessed invidually and changes from the following scoring (within the task) my occur depending on what the remaining answer looks like.

Question 1 [5p]

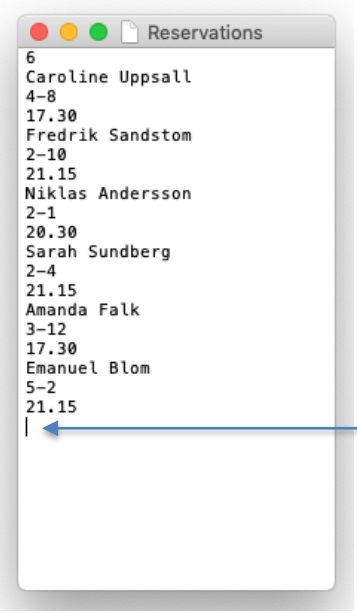
What will the print out look like when the following code is run with the input ”Salmon sushie dish”?

```
#include <stdio.h>
#include <string.h>
#define STRMAX 30

int main(void)
{
    char str[STRMAX], strNew[STRMAX*2];
    int i;
    printf("Enter a string: ");
    fgets(str, STRMAX, stdin);
    str[strlen(str)-1] = '\0';

    for(i = 0; str[i] != '\0'; i++)
    {
        strNew[i] = str[i];
        if(strNew[i] == 's' || strNew[i] == 'S')
            strNew[++i] = 's';
    }
    strNew[i] = '\0';

    puts(strNew);

    return 0;
}
```

Answer:
Sslmon ssssie diss

Question 2 [14p]

Suppose you have the following file (called Reservations) containing the reservations made for a day at a restaurant. You will be working with this information in different ways in the following sub-questions.



This is the text cursor to show that there is an enter character at the end of the file. Oterwise, there are no hidden characters in the file. However there might be characters that are vivisble but still easily missed.

On the first line of tje file you have the number of reservations written down in the file. Each reservation is then described with the following information:

Name of the person who made the reservation.

Number of people in the dinner company – table number

Time for the reservation.

The first reservation in the file is made by Caroline Uppsall, there are four people in the dinner company and they have been given table number 8. The time of the reservation is 17.30.

a) Create a structure type that can hold the information about a reservation contained in the file above. The structure should be called ”struct Reservation”. [2p]

Solution example (2p)

```
struct Reservation
{
    char name[NAMEMAX];
    int nrOfPeople
    int tableNr;
    float time
};
```

b) Open the file (Reservations) and read all the information in the file (about the varius reservations) and save it in a dymanically allocated array. The pointer to the dynamic array should be of the type you created in a) and called todaysReservations. [7p]

Solution example (7p)

```
struct Reservation *todaysReservations;
int nrOfReservations;
FILE *fp = fopen("/Users/cul01/Desktop/Reservations", "r"); (1p)
if(fp != NULL)    (0.5p)
{
    fscanf(fp, "%d\n", &nrOfReservations); (0.5p)
    todaysReservations = (struct Reservation*)calloc(nrOfReservations,
                                                     sizeof(struct Reservation)); (1p)
    if(todaysReservations != NULL) (0.5p)
    {
        for(int i = 0; i < nrOfReservations; i++) (1p)
        {
            fgets(todaysReservations[i].name, NAMEMAX, fp); (0.5p)
            //todaysReservations[i].name[strlen(todaysReservations[i].name)-1] = '\0';
            fscanf(fp, "%d-%d\n", &todaysReservations[i].nrOfPeople,
                                  &todaysReservations[i].tableNr); (1p)
            fscanf(fp, "%f\n", &todaysReservations[i].time); (0.5p)
        }
    }
    fclose(fp); (0.5p)
```

c) Write a line of code that changes the table number of the reservation on index 2 (Niklas Andersson) to 3. You must use the arraow operator (->). You should change the information saved in the memory in the program (task b), not the file. If you havn't completed task b you can still do this task. [1p]

Solution example (1p)

```
(todaysReservations+2)->tableNr = 3;
```

d) Create a function that writes the entire list of reservations binary to a file. You should use the following function declaration: [4p]

```
void saveToBinaryFile(struct Reservation *allReservations, int nrOfReservations, char *filename)
```

The entire array of reservations should be sent to the parameter allReservations. For the parameter noOfReservations the number of reservations in the array should be sent and for filename, the file name that the user wants for the create binary file should be sent.

You can do this task even if you havn't done task b that reads the information from the text file.

Solution example (4p)

```
void saveToBinaryFile(struct Reservation *allReservations, int nrOfReservations,
                                                                    char *filename)
{
    FILE *fp = fopen(filename, "wb"); (1p)
    if(fp != NULL) (0.5p)
    {
        fwrite(&nrOfReservations, 1, sizeof(int), fp); (1p)
        fwrite(allReservations, nrOfReservations, sizeof(struct Reservation),  fp); (1p)
    }
    fclose(fp); (0.5p)
}
```

Question 3 [5p]

What does the print out look like when the program is finished?

```c
#include <stdio.h>

int main(void)
{
    int a = 1, b = 2, i = 3;
    int arr[] = {5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60};
    int *p = &b, *pArr = arr;
    while (i < 10) {
        a = *pArr;
        pArr++;
        if (i < 4 || i == 8)
        {
            i += 3;
            p = &b;
        }
        else
        {
            i++;
            p = &a;
            (*p)++;
        }
        *p = *p + 1;
    }
    printf("a = %d\nb = %d\ni = %d\n", a, b, i);
    return 0;
}
```

Answer:

a = 20

b = 4

i = 11

Question 4 [5p]

Which of the following statements is not true? Your are allowed to give a maximum of 10 answers, if you give more anwers than 10 only the first 10 will be assessed.

1) In a variable declared with int, both negative and positive integers can be saved
2) * is used fo find out the address where a variable is located in the memory
3) Dynamically allocated memory exists until it is released with the free function
4) The function strlen returns the size of a string up until (but not including) the string end characted '\0'
5) A pointer variable does not take up any memory
6) A function can be called several times in a program
7) malloc or calloc is used to allocate memory on the heap
8) The char type can only be used to create strings
9) If you want to compare two strings, strcmp must be used
10) fgets requires that the stdlib library is included
11) A pointer variabel contains an address
12) Modulu (%) is used to divide two integers
13) A function must always return something
14) The do-while loop is preferred to use if the body of the loop should be run at least once
15) A loop body is run if the loop condition evaluates to true
16) To compare two values, use =
17) If arr is and declared array, *(arr+i) is the same as arr[i]
18) An array must be of char-type
19) A structure (posts) is always allocated on the heap

Question 5 [2p]

What will the following expressions evaluate to (true/1 or false/0). You have the following variables declared and assigned:

```
int a = 9, b = 4, x = 3, y = 13;
```

a)
```
(a % 2 == 0 && a/b <= x) || (y - a < b)
```

b)
```
a/2 > b && !((y - b > a) && a % x == 0)
```

Answer:

```
a) False (0)
b) False (0)
```

Question 6 [4p]

For the following function call, specify the function head(ie what is used when defining and declaring the function).

Assume the following type definition:

```
struct Car{
    char brand[10];
    int productionYear;
    float distance;
};
```

Assume that the follwing declarations are in the program. You can assume that all variables used in are set to valid values.

```
struct Car myCar, *anotherCar;
float x, *y;
int array[10], z;
char letter;
```

Example:

Function call:	Give the declaration for
`x = func(6.3, letter);`	`func()`

Answer: `float func(float x, char c);`

Explanation: Since func() takes a float value and a characted as arguments, the parameters be of the float and char types (what they are called is less important). The return value is assigned to x which is declared as a float, therefore func() must have a float as the return type.

Task:

Function call:	Give the declaration for
1) `anotherCar = func1(10);`	`func1()`
2) `func2(&x, array);`	`func2()`
3) `letter = func3(*(array+1), myCar.distance);`	`func3()`
4) `anotherCar->productionYear = func4(myCar.brand, 'a');`	`func4()`

Lösning

```
1) struct Car *func1(int a);
2) void func2(float *r, int arr[]);
3) char func3(int z, float x);
4) int func4(char *str, char l);
```

Question 7 [8p]

Write a program that allows the user to enter integers that you will use to calculate a result in the following way (the result should start counting from 0).

- If the number is even, the number should be subtracted from the result.
- If the number is odd, the number should be added to the result.
- If the number is negative, it should be converted to a positive number and then the rest of the rules should be applied (-13 becomes 13, -7 becomes 7, -98 becomes 98 etc).
- If the number is evenly divisible by 10, a number between 1 and 50 should be randomly generated and then added to the result.
- If the number is greater than 100, it should be ignored.
- If the number is 0, end the inputs and print the result on the screen.

Example run

```
Enter a number (0 to exit): 5
Enter a number (0 to exit): 20
Enter a number (0 to exit): 24
Enter a number (0 to exit): 104
Enter a number (0 to exit): -19
Enter a number (0 to exit): 10
Enter a number (0 to exit): 0

The result is 63
```

The calculation was made as follows:
0+5 → 5
5+23 → 28 (23 is random since 20 is evenly divisible by 10)
28-24 →4
104 is ignored
4+19 → 23 (-19 is converted to 19)
23 + 40 → 63 (40 is random since 10 is evenly divisible by 10)

Solution example:

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(void)
{
    int number, result = 0; //(0.5p)
    srand(time(0));
    do{
        printf("Enter a number (0 to exit): ");
        scanf("%d", &number); //(0.5p)
        if(number < 0) //(0.5p)
            number *= -1; //(0.5p)
        if(number != 0 && number <= 100) //(1p)
        {
            if(number%10 == 0) //(0.5p)
                result += rand()%50+1; //(1p)
            else if(number%2 == 0) //(0.5p)
                result -= number; //(0.5p)
            else if(number%2 == 1) //(0.5p)
                result += number; //(0.5p)
        }
    }while(number != 0); //(1p)

    printf("The result is %d\n", result); //(0.5p)
    return 0;
}
```