Exercise 3: Conditional and Alternative statements

R Ram Kaushik

20-02-18

1 24 hour format

Write a program to get a time in 24 hour format and convert it to a 12 hour format

1.1 Program Design

The program consists of `main()`, which gets the input of time from `stdin`, converts it to 12 hour format and prints the result on `stdout`.

1.2 Algorithm

def func(h,m,s): if h<=12: if h==12: print("%d:%d:%d pm",h,m,s) else print("%d:%d:%d am",h,m,s) else if h==24: print("%d:%d:%d am",h-24,m,s) else print("%d:%d:%d pm",h-12,m,s)

1.3 Source Code:

```
#include<stdio.h>
int main(){
  int h,m,s;
  scanf("%d%d%d",&h,&m,&s);
  if(h<=12){
    if(h==12{
      printf("%d:%d:%d pm\n",h,m,s);
    }
    else{
      printf("%d:%d:%d am\n",h,m,s);
    }
  }
  else{
    if(h==24){
      printf("%d:%d:%d am\n",h-24,m,s);
```

```
    }
    else{
      printf("%d:%d:%d pm\n",h-12,m,s);
    }
  }

}
```

1.4 Test Input:

```
15 34 23
```

1.5 Output:

```
3:34:23 pm
```

2 Time Comparison

Write a function to accept 2 time in hours minutes and seconds and compare which time is earlier.

2.1 Program Design

The program consists of `main()`, which gets the input from `stdin`, compares the times and prints the result on `stdout`.

2.2 Algorithm

```
def func(h1,m1,s1,h2,m2,s2):
  if h1>h2:
    print("t1 is earlier")
  elif h1<h2:
    print("t2 is earlier")
  else
    if m1>m2:
      print("t1 is earlier")
    elif m1<m2:
      print("t2 is earlier")
    else:
      if s1>s2:
        print("t1 is earlier")
```

```
      elif s1<s2:
        print("t2 is earlier")
      else:
        print("Both are same")
```

2.3 Source Code:

```
#include<stdio.h>
int main()
{
  int h1,m1,s1,h2,m2,s2;
  scanf("%d%d%d",&h1,&m1,&s1);
  scanf("%d%d%d",&h2,&m2,&s2);
  printf("%d %d %d\t%d %d %d\n",h1,m1,s1,h2,m2,s2);
  if(h1<h2){
    printf("t1 is earlier");
  }
  else if(h1>h2){
    printf("t2 is earlier");
  }
  else{
    if(m1>m2){
      printf("t2 is earlier");
    }
    else if(m1<m2){
      printf("t1 is earlier");
    }
    else{
      if(s1>s2){
printf("t2 is earlier");
      }
      else if(s1<s2){
printf("t1 is gtreater");
      }
      else{
printf("both are equal");
      }
    }
  }
```

```
}
```

2.4 Test Input:

```
16 23 45  13 23 43
```

2.5 Output:

```
t2 is earlier
```

3 Time difference

Write a program to calculate the time difference between the two time the user enters and print it

3.1 Specification

A function `sign()`, which takes an integer as the input and returns it's sign to the calling function.

3.2 Prototype

```
int sign(int a);
```

3.3 Program Design

The program consists a function `sign(int a)`, which returns the sign of the integer, and `main()`, which gets the input from `stdin`, calls the function and prints the rsult accordingly on `stdout`.

3.4 Algorithm

```
def sign(a):
  if a>=0:
    return 1
  else
    return -1
```

3.5 Sorce Code

```
#include<stdio.h>

int sign(int a){
  if(a>=0){
    return 1;
  }
  else{
    return -1;
  }
}

int main(){
  int a,b,c,d,e,f,g,h,i;
  scanf("%d%d%d",&a,&b,&c);
  scanf("%d%d%d",&d,&e,&f);
  g=sign(a-d);
  h=sign(b-e);
  i=sign(c-f);
  if(g>0){
    if(h>0 && i>0){
      printf("%d:%d:%d\n",a-d,b-e,c-f);
    }
    else if(h>0 && i<0){
      printf("%d:%d:%d\n",a-d,b-e,f-c);
    }
    else if(h<0 && i>0){
      printf("%d:%d:%d\n",a-d,e-b,c-f);
    }
    else{
      printf("%d:%d:%d\n",a-d,e-b,f-c);
    }
  }
  else{
    if(h>0 && i>0){
      printf("%d:%d:%d\n",d-a,b-e,c-f);
    }
    else if(h>0 && i<0){
```

```
      printf("%d:%d:%d\n",d-a,b-e,f-c);
    }
    else if(h<0 && i>0){
      printf("%d:%d:%d\n",d-a,e-b,c-f);
    }
    else{
      printf("%d:%d:%d\n",d-a,e-b,f-c);
    }
  }
}
```

3.6 Test Input

```
18 16 24        13 15 23
```

3.7 Output

```
5 1 1
```

4 Smallest and largest of 4 numbers

Write a program to find the smallest and largest number out of the 4 numbers entered from the standard input

4.1 Specification

2 functions `min2()` and `max2()`, which take 2 integers as the input and returns the minimum and maximum of the two to the calling function respectively.

4.2 Prototype

```
int min2(int a, int b);
int max2(int a, int b);
```

4.3 Program Design

The program consists of 2 functions `min2(int a, int b)` and `max2(int a, int b)` which returns the minimum and maximum of the 2 numbers, and `main()`, which gets the input from `stdin`, calls the functions, and prints the result on `stdout`.

4.4 Algorithm

```
def min2(a,b):
  if a>b:
    return b
  else:
    return a
def max2(a,b):
  if a<b:
    return b
  else:
    return a
```

4.5 Source Code

```
#include<stdio.h>
int min2(int a, int b){
  if(a>b){
    return b;
  }
  else{
    return a;
  }
}

int max2(int a, int b){
  if(a<b){
    return b;
  }
  else{
    return a;
  }
}

int main(){
  int a,b,c,d,m,n;
  scanf("%d%d%d%d",&a, &b, &c, &d);
  m=min2(a,b);
  m=min2(m,c);
  m=min2(m,d);
```

```
  n=max2(a,b);
  n=max2(n,c);
  n=max2(n,d);
  printf("%d,%d\n",m,n);
}
```

4.6 Test Input

```
23 43 65 12
```

4.7 Output

```
12 65
```

5 Grades

Write a function `grades()` to translate the marks of a student in various subjects into letter grades and print the grades on the output.

Mark range	Grade points	Leter grade
91-100	10	S
81-90	9	A
71-80	8	B
61-70	7	C
57-60	6	D
51-56	5	E
<50	0	U

5.1 Specification

A function `grade()`, which gets the mark as the input and returns a grade as character to the calling function.

5.2 Prototype

```
char grade(int x);
```

5.3 Program Design

The program consists of a function `grade(int x)`, which returns a grade as a character based on the mark, and `main()`, which gets the input from `stdin`, calls the function and prints the result on `stdout`.

5.4 Algorithm

```
def grade(x):
  if x>90:
    return 's'
  elif x>80:
    return 'a'
  elif x>70:
    return 'b'
  elif x>60:
    return 'c'
  elif x>56:
    return 'd'
  elif x>50:
    return 'e'
  else:
    return 'u'
```

5.5 Source Code

```
#include<stdio.h>
char grade(int x){
  if(x>90){
    return 's';
  }
  else if(x>80){
    return 'a';
  }
  else if(x>70){
    return 'b';
  }
  else if(x>60){
    return 'c';
  }
  else if(x>56){
    return 'd';
  }
  else if(x>50){
    return 'e';
  }
```

```
  else{
    return 'u';
  }
}
int main(){
  int a[20],n;
  char g;
  scanf("%d",&n);
  for(int i=0;i<n;i++){
    scanf("%d",&a[i]);
  }
  for(int i=0;i<n;i++){
    g=grade(a[i]);
    printf("%c\n",g);
  }
}
```

5.6 Test Input

```
8
100 98 78 45 98 78 40 90
```

5.7 Output

```
s
s
b
u
s
b
u
a
```

6 Tariff Calculator

Write a function `eb()` to find out the domestic eb bill based on the given slab rates

1. Consumption upto 100 units: free.

2. Consumption above 100 units and upto 200 units: Rs 1.50 per unit.
3. Consumption above 200 units and upto 500 units: Rs 2.00 per unit for 101-200 units and Rs 3.00 per unit for 201-500 units.
4. Consumption above 500 units: Rs 3.50 per unit for 101-200 units, Rs 4.60 per unit for 201-500 units, and Rs 6.60 beyond 500 units.

6.1 Specification

A function `eb()`, which takes the number of units as the input and returns the cost based on the conditions to the calling function.

6.2 Prototype

```
float eb(int unit);
```

6.3 Program Design

The program consists of a function `eb(int unit)`, which returns the net cost, and `main()`, which gets the input from `stdin`, calls the function and prints the result on `stdout`.

6.4 Algorithm

```
def eb(u):
  if u<=100:
    return 0
  elif u>100 and u<=200:
    return 1.5*u
  elif u>200 and u<=500:
    return (u-200)*3.0+(u-100)*2.0
  else:
    return (u-500)*6.6+(u-200)*4.6+(u-100)*3.5
```

6.5 Source Code

```
#include<stdio.h>
float eb(int unit){
  if(unit<=100){
    return 0.0;
  }
  else if((unit>100)&&(unit<=200)){
```

```
    return 1.5*unit;
  }
  else if((unit>200)&&(unit<=500)){
    return(unit-200)*3.0+100*2.0;
  }
  else{
    return (unit-500)*6.6+300*4.6+100*3.5;
  }
}
int main(){
  int unit;
  float cost;
  scanf("%d",&unit);
  cost=eb(unit);
  printf("%.4f\n",cost);
}
```

6.6 Test Input

```
700
```

6.7 Output

```
3050.0000
```

7 Income Tax

Write a function `tax()` to calculate the income tax based on the age and the income of the person

1. Income Tax Slab for Individual Tax Payers (Less Than 60 Years Old)

Income Slab	Tax Rate
Up to Rs.2,50,000	No tax
Rs.2,50,000 - Rs.5,00,000	5%
Rs.5,00,000 - Rs.10,00,000	20%
Rs.10,00,000 and beyond	30%

1. Income Tax Slab for Senior Citizens (60 Years Old Or more but Less than 80 Years Old)

Income Slab	Tax Rate
Up to Rs.3,00,000	No tax
Rs.3,00,000 - Rs.5,00,000	5%
Rs.5,00,000 - Rs.10,00,000	20%
Rs.10,00,000 and beyond	30%

1. Income Tax Slab for Senior Citizens (More than 80 years old)

Income Slab	Tax Rate
Up to Rs.2,50,000	No tax
Rs.2,50,000 - Rs.5,00,000	No tax
Rs.5,00,000 - Rs.10,00,000	20%
Rs.10,00,000 and beyond	30%

Modify your function to take the age and the income as the parameters and calculate the tax.

7.1 Specification

A function `tax()`, which gets the age and income as the inputs, checks the conditions and returns the value of tax to the calling function

7.2 Prototype

```
float tax(int age, int income);
```

7.3 Program Design

The program consists of a function `tax(int age, int income)`, which returns the value of tax based on conditions, and `main()`, which gets the input from `stdin`, calls the function and prints the result on `stdout`.

7.4 Algorithm

```
def tax(age,income):
  if(age<60):
      if income<250000:
        return 0.0
      elif income>=250000 and income<500000:
        return (5.0/100)*income
      elif income>=500000 andincome<1000000:
        return (20.0/100)*income
      else:
        return (30.0/100)*income
    else ifage>=60 and age<80:
```

```
    if income<300000;
      return 0.0
    elif income>=300000 and income<500000:
      return (5.0/100)*income
    elif income>=500000 and income<1000000:
      return (20.0/100)*income
    else:
      return (30.0/100)*income
  else:
    if income<500000:
      return 0.0
    elif income>=500000 and income<1000000:
      return (20.0/100)*income
    else:
      return(30.0/100)*income
```

7.5 Source Code

```
#include<stdio.h>
float tax(int age, int income){
  if(age<60){
    if(income<250000){
      return 0.0;
    }
    else if((income>=250000)&&(income<500000)){
      return (5.0/100)*income;
    }
    else if((income>=500000)&&(income<1000000)){
      return (20.0/100)*income;
    }
    else{
      return (30.0/100)*income;
    }
  }
  else if((age>=60)&&(age<80)){
    if(income<300000){
      return 0.0;
    }
    else if((income>=300000)&&(income<500000)){
```

```
      return (5.0/100)*income;
    }
    else if((income>=500000)&&(income<1000000)){
      return (20.0/100)*income;
    }
    else{
      return (30.0/100)*income;
    }
  }
  else{
    if(income<500000){
      return 0.0;
    }
    else if((income>=500000)&&(income<1000000)){
      return (20.0/100)*income;
    }
    else{
      return(30.0/100)*income;
    }
  }
}
int main()
{
  int age,income;
  float t;
  scanf("%d%d",&age,&income);
  t=tax(age,income);
  printf("%f\n",tax);
}
```

7.6 Test Input

```
85 3000000
```

7.7 Output

```
900000.0
```

8 Inversion

In a sequence of integers `a0, a1, a2, a3`, any pair of integers `(ai, aj)` is said to be an *inversion* if `ai > aj` for `i < j`. Write a program to correct/order all the inversions in the sequence.

8.1 Specification

A function `inversion()`, which takes an array and it's length as input, counts the number of inversions to be performed and returns the result to the calling function.

8.2 Prototype

```
int inversion(int a[], int n);
```

8.3 Program Design

The program consists of a function `inversion(int a[], int n)`, which counts the number of inversions to be done, and `main()`, which gets the input from `stdin`, calls the function, and prints the result on `stdout`.

8.4 Algorithm

```
def inversion(a,n):
  c=0
  for i in range(n):
    for j in range(i+1,n):
      if a[i]>a[j]:
        c+=1
  return c
```

8.5 Source Code

```
#include<stdio.h>
int inversion(int a[], int n){
  int c=0;
  for(int i=0;i<n;i++){
    for(int j=i+1;j<n;j++){
      if(a[i]>a[j]){
c++;
      }
    }
  }
```

```
  return c;
}
int main(){
  int a[20],n;
  scanf("%d",&n);
  for(int i=0;i<n;i++){
    scanf("%d",&a[i]);
  }
  int c=inversion(a,n);
  printf("%d",c);
}
```

8.6 Test Input

```
5
1 20 6 4 5
```

8.7 Output

```
5
```