

به نام پروردگار قلم

فهرست مطالب:

۲ ایده اصلی:

۴ توضیحات بخش های برنامه:

۷ تصاویر چند اجرا:

۸ کد نهایی:

ایده اصلی:

طبق توضیحات پروژه، باید برنامه ای نوشت که فراوانی تعدادی داده از جنس اعداد (integer) را محاسبه کند. مطابق ورودی پروژه اعداد ورودی باید به این گونه باشند:

17, 18, 24, 3, 6, 7, 7, 8, 9, 17, 17, 12, 17

یعنی با کما (,) از یکدیگر جدا شوند. برای اینکه بتوانیم از کاربر چنین رشته ای را دریافت کنیم باید از رشته (string) که به نوعی آرایه ای است که از جنس کاراکتر (character) می باشد باید استفاده کنیم چرا که کما از جنس اعداد نیست.

بدین منظور ما باید ابتدا یک رشته تعریف کرده و سپس به کاربر بگوییم اعداد مورد نظر خود را وارد کرده و با کما از هم جدای شان کند. بعد از اینکه کاربر اعداد خود را به شیوه گفته شده وارد کرد آن را در رشته ای که قبلاً تعریف کرده ایم ذخیره می کنیم.

حال با تعریف یک حلقه و دو متغیر تعداد اعداد وارد شده را محاسبه می کنیم و در یکی از متغیرها ذخیره میکنیم و آرایه ای به طول آن متغیر می سازیم. در این پروژه به این دلیل که اعداد را با کما جدا کردیم تعداد اعداد برابر با یک به علاوه تعداد کما است.

حالا باید اعداد وارد شده در رشته که از جنس کاراکتر هستند را به اعداد integer تبدیل کنیم. برای رسیدن به این هدف باید این نکته را بدانیم که هر کاراکتر کد اختصاصی¹ ASCII مخصوص به خود را دارد و کامپایلر ها (compiler) هر کاراکتر را با این کد شناسایی میکنند، حالا با دانستن این موضوع ما باید این را هم بدانیم که کاراکتر صفر با عدد صفر برابر نیست و مقدار کاراکتر صفر با عدد 48 در مبنای دهی دهی برابر است. در نتیجه ما میتوانیم با تعریف یک تابع کاراکتر را به عدد تبدیل کنیم و باید در انجام این امر به نکته بالا نیز توجه کنیم.

بعد از تعریف تابع بالا و انتخاب یک نام مناسب برای آن ما به یک حلقه که از آن برای خواندن اعداد کاراکتری و تبدیل آن به اعداد integer و ذخیره اعداد integer در یک آرایه که قبلاً آن را تعریف کرده ایم، نیاز داریم.

در گام بعد از یک حلقه برای مرتب کردن اعداد آرایه استفاده میکنیم و داده ها را مرتب می کنیم. می توان این کار را با مقایسه کردن ترتیبی داده های آرایه انجام داد. در نهایت اعداد به ترتیب و پشت سر هم قرار می گیرند.

حال به محاسبه فراوانی می پردازیم. یک آرایه و دو حلقه ی تو در تو با سه متغیر تعریف میکنیم. متغیر اول را به عددی که قرار است با بقیه اعداد مقایسه شود اختصاص میدهیم، متغیر دوم را برای عددی که در حلقه دوم برای مقایسه عدد با دیگر اعداد آرایه است، استفاده می کنیم، متغیر سوم را طوری تعریف میکنیم که هرگاه هنگام مقایسه اعداد برابر پیدا کردیم افزایش یابد و هنگامی که مقایسه تمام شده در خانه ای با مقدار متغیر در حال مقایسه ذخیره شود.

تا اینجا ما دو آرایه داریم یک آرایه که اعداد در آن ذخیره شده اند و آرایه دیگر فراوانی همان داده ها در خانه ای برابر با خانه آرایه اول ذخیره شده اند.

اگر همینجا تصمیم به چاپ اعداد و مقدار فراوانی شان بگیریم به یک مشکل میخوریم و آن این است که اعداد به تعداد تکرارشان چاپ و مقدار فراوانی شان نشان داده می شود.

برای حل این مشکل ما باید تابعی بنویسیم که داده های تکراری را حذف کند (چرا که دیگر به آنها نیاز نداریم و فراوانی اعداد را محاسبه کرده ایم.) و چون از قبل داده ها را مرتب کرده ایم این تابع نیز بر اساس مقایسه خانه های آرایه با خانه های بعدی شان می تواند کار کند. پس این مشکل بر طرف میشود.

حالا وقت کد کردن ایده است.

در ادامه بخش های مختلف برنامه را توضیح می دهیم.

توضیحات بخش های برنامه:

در بخش اول کتابخانه های مورد نیاز و توابع استفاده شده را معرفی کرده ام. در این برنامه از کتابخانه string استفاده کرده ام.

تابع ($stoi^2$): این تابع کاراکتر را از مقدار کاراکتر 0 که مقدار 48 را دارد کم میکند. اینکار باعث می شود کاراکتر به عدد تبدیل شود.

تعریف متغیر ها : در بخش اول تابع اصلی شروع به تعریف متغیر کرده ام. متغیر های i, j, l, n, z برای گام ها، متغیر l برای شمارش تعداد اعداد وارد شده و متغیر n برای ذخیره اعداد تبدیل شده از رشته به integer تعریف شده اند.

در ادامه رشته nums برای دریافت اعداد از کاربر تعریف شده است و با نمایش پیام خط سوم شکل در خط چهارم این رشته دریافت میشود.

متغیر len برای ذخیره طول رشته nums تعریف شده است. در این متغیر با استفاده از تابع $length()$ که در کتابخانه string موجود است طول رشته وارد شده توسط کاربر محاسبه و در متغیر len ذخیره میشود.

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 int stoi (char c) { return c - '0'; }
5 int main(){
```

```
1 int i=0,j=0,l=1,n=0,z=0;
2 string nums;
3 cout << "Please Enter Number And Split Them With ',' \n";
4 cin >> nums;
5 int len= nums.length();
```

```

1  or(i=0;i<len;i++){//bugs_hunt here
2      if(nums[i+1]==' '){
3          cout << "Please Enter Number Correctly!\nLike:\n1,2,3,4\nYou Have Space ' ' Between Your Numbers!\n";
4          system("pause");
5          return 0;
6      }
7      if (nums [0]==' '){
8          cout << "Please Enter Number Correctly!\nLike:\n1,2,3,4\nYou Have ' ' Insted of first number!\n";
9          system("pause");
10         return 0;
11     }
12     if (nums[i]>='0'&&nums[i]<='9'&&nums[i+1]!='-'){
13         cout << "Please Enter Number Correctly!\nLike:\n1,2,3,4\nYou Have Something Like This '1-1' Between Your Number!\n";
14         system("pause");
15         return 0;
16     }
17     if (nums[i]!='-'&&nums[i+1]!='-'){
18         cout << "Please Enter Number Correctly!\nLike:\n1,2,3,4\nYou Have '-' Between Your Number!\n";
19         system("pause");
20         return 0;
21     }
22     if (nums[i]==','&&nums[i+1]==',' ){
23         cout << "Please Enter Number Correctly!\nLike:\n1,2,3,4\nYou Have ',' Between Your Number!\n";
24         system("pause");
25         return 0;
26     }
27     if (! (nums[i]>='0'&&nums[i]<='9' || nums[i]=='-' || nums[i]!='-')){
28         cout << "Please Enter Number Correctly!\nLike:\n1,2,3,4\nYou Enter Incorrect Charcter Like:\n'a,A,~,etc'\n";
29         system("pause");
30         return 0;
31     }
32     if (nums[i]==','&&nums[i+1]=='\0'){
33         cout << "Please Enter Number Correctly!\nLike:\n1,2,3,4\nYou Have Something Like ' , ' Between In End Of Your Number\nReenter Your Number And Dont Enter ' , ' At End\n";
34         system("pause");
35         return 0;
36     }
37 }

```

رفع برخی از مشکلات : در این قسمت برای جلوگیری از اشکال در اجرای برنامه برای چندین مشکل پیغام و واکنش مناسب تعریف شده که در تصویر مشخص است.

محاسبه تعداد اعداد و تعریف آرایه : در این حلقه ی ساده تعداد اعداد با استفاده از تعداد کما ها محاسبه میشود.

در پایان تعداد اعداد در متغیر l ذخیره شده و آرایه num را با طول l تعریف کرده ام.

در اینجا متغیر isNegative نیز از نوع بولین (Boolean) و با ارزش نادرست تعریف شده است که در حلقه بعدی از آن استفاده میکنم.

```

1  for(i=1;i<=len;i++){//count number
2      if(nums[i]==' '){
3          l++;
4      }
5  }
6  int num[l];
7  bool isNegative = false;

```

تبدیل اعداد کاراکتری به integer و ذخیره آن در آرایه: در این حلقه بعد از مقدار دهی و تعیین شرط در خط 3 تصویر شرط منفی بودن عدد وارد شده بررسی شده و در صورت منفی بود مقدار متغیر isNegative به درست تغییر می کند و دوباره حلقه را اجرا میکند. در شرط بعد اگر در رشته nums کما رسیدیم،

```

1  for (z=0,i=0;i<len;i++){//this cycle convert string to number and save number to N[] array
2      {
3          if (nums[i] == '-') {
4              isNegative = true;
5              continue;
6          }
7
8          if (nums[i] == ',') {
9              if (isNegative)
10                 j = -j;
11                 num[z++] = j;
12                 j = 0;
13                 isNegative = false;
14                 continue;
15             }
16
17             n = stoi(nums[i]);
18             j = j * 10 + n;
19
20             if (i == len - 1) {
21                 if (isNegative)
22                     j = -j;
23                 num[z] = j;
24             }
25         }
26     }

```

شرط منفی بودن را بررسی در صورت منفی بودن عدد را که در متغیر `j` ذخیره شده قرینه میکند و دوباره در متغیر `j` ذخیره می کند. در ادامه عدد را در خانه بعدی آرایه ذخیره میکند و متغیر `j` را صفر و ارزش متغیر `isnegative` را به غلط تغییر میدهد و تابع را دوباره اجرا میکند.

در خط بعد مقدار متغیر `n` را برابر با مقدار `integer` کاراکتر عدد درون رشته `nums` قرار می دهیم سپس در خط بعدی ارقام به یک مقدار دهی تبدیل میشوند.

در خط 20 اگر مقدار گام ما یک واحد کمتر از طول رشته بود شرط منفی بودن را بررسی و سپس عدد را در آرایه `num` ذخیره میکند.

مرتب کردن اعداد آرایه از کوچک به بزرگ:

```
1 for (int i = 0; i < l - 1; i++)//number sort!
2     for (int j = 0; j < l - i - 1; j++)
3         if (num[j] > num[j + 1]) {
4             int temp = num[j];
5             num[j] = num[j + 1];
6             num[j + 1] = temp;
7         }
```

در این قسمت از دو حلقه به صورت تو در تو استفاده کرده ام تا اعداد را از کوچک به بزرگ مرتب کند.

طریقه کار به این صورت است که عدد اول آرایه را با تمام اعداد آرایه به صورت متوالی مقایسه می کند. یعنی اگر عدد در خانه اول از عدد درون خانه دوم

بزرگ تر باشد عدد خانه اول را در متغیر `temp` ذخیره و سپس عدد خانه دوم را به خانه اول انتقال میدهد و عدد ذخیره شده درون متغیر `temp` (عدد خانه اول) را به خانه دوم انتقال میدهد و این روند تا آخرین عدد درون آرایه ادامه می یابد.

```
1 int f[1];
2 for(i=0;i<l;i++){// frequncy calculate here
3     for(z=0,j=0;j<l;j++)
4         if(num[i]==num[j])
5             z++;
6     f[i]=z;
7     z=0;
8 }
```

محاسبه فراوانی اعداد: ابتدا آرایه `f` را به طول 1 تعریف کرده ام تا در آن فراوانی اعداد درون آرایه `num` را با همان ترتیب و اندازه ذخیره کنم.

در اینجا هم از دو حلقه به صورت تو در تو استفاده کرده ام به این صورت که عدد اول آرایه `num` را با همه خانه های آرایه فوق مقایسه و در صورتی که عددی به فرض در خانه اول با خانه

سوم برابر بود، متغیر `z` را یک واحد اضافه می کند و در نهایت مقدار متغیر `z` که همان فراوانی است را در خانه ای برابر با خانه آرایه `num` ذخیره می کند و در نهایت مقدار متغیر `z` را برای استفاده دوباره صفر میکند.

نمایش اعداد و فراوانی شان:

```

1  for (i=0;i<l;i++){//printer
2      if(num[i]==num[i+1])
3          continue;
4      cout << "F["<<num[i]<<"]="<<f[i]<<endl;}}
5  system("pause");
6  return 0;
7  }
```

در حلقه درون تصویر پس از مقدار دهی اولیه و تعیین شرط، شرط خط 2 را بررسی میشود.

شرط از این قرار است که اگر خانه i با خانه $i+1$ برابر بود دوباره حلقه را اجرا کن. دلیل این شرط گذاری این است که از نمایش اعداد تکراری و فراوانی آنها جلوگیری شود. این

شرط هنگامی به درستی کار میکند که اعداد درون آرایه مرتب شده باشند که ما اینکار را در قسمت های قبل انجام داده ایم.

در ادامه پس از بررسی شرط فوق اعداد با مقدار فراوانی شان به الگوی زیر به نمایش در می آیند.

فراوانی عدد = $F[\text{عدد}]$

کد خط 5 تصویر باعث میشود برنامه compile شود پس از انجام تابع به طور ناگهانی بسته نشود. پیام:

Press any key to continue . . .

و فشردن هر یک از کلید های صفحه کلید بسته میشود.

تصاویر چند اجرا:

```

Please Enter Number And Split Them With ',':
1,1,1
F[1]=3
Press any key to continue . . .
```

```

Please Enter Number And Split Them With ',':
17,18,24,3,6,7,7,8,9,17,12,17
F[3]=1
F[6]=1
F[7]=2
F[8]=1
F[9]=1
F[12]=1
F[17]=4
F[18]=1
F[24]=1
Press any key to continue . . . |
```

```

Please Enter Number And Split Them With ',':
1234567890,1234567890,-1234567890,-1234567890,-100,-1000,-100,1,2,3,1,2,3
F[-1234567890]=2
F[-1000]=1
F[-100]=2
F[1]=2
F[2]=2
F[3]=2
F[1234567890]=2
Press any key to continue . . . |
```

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int stoi (char c) { return c - '0'; }
5  int main(){
6      int i=0,j=0,l=1,n=0,z=0;
7      string nums;
8      cout << "Please Enter Number And Split Them With ',' : \n";
9      cin >> nums;
10     int len= nums.length();
11     for(i=0;i<len;i++){//bugs hunt here
12         if(nums[i+1]==' '){
13             cout << "Please Enter Number Correctly!\nLike:\n1,2,3,4\nYou Have Space ' ' Between Your Numbers!\n";
14             system("pause");
15             return 0;
16         }
17         if (nums [0]==','){
18             cout << "Please Enter Number Correctly!\nLike:\n1,2,3,4\nYou Have ',' Insted of first number!\n";
19             system("pause");
20             return 0;
21         }
22         if (nums[i]>='0'&&nums[i]<='9'&&nums[i+1]=='-'){
23             cout << "Please Enter Number Correctly!\nLike:\n1,2,3,4\nYou Have Something Like This '1-1' Between Your Number!\n";
24             system("pause");
25             return 0;
26         }
27         if (nums[i]=='-'&&nums[i+1]=='-'){
28             cout << "Please Enter Number Correctly!\nLike:\n1,2,3,4\nYou Have '--' Between Your Number!\n";
29             system("pause");
30             return 0;
31         }
32         if (nums[i]==','&&nums[i+1]==','){
33             cout << "Please Enter Number Correctly!\nLike:\n1,2,3,4\nYou Have ',,' Between Your Number!\n";
34             system("pause");
35             return 0;
36         }
37         if (!(nums[i]>='0'&&nums[i]<='9' || nums[i]=='-' || nums[i]=='-')){
38             cout << "Please Enter Number Correctly!\nLike:\n1,2,3,4\nYou Enter Incorrect Charcter Like:\n'a,A,~,etc'\n";
39             system("pause");
40             return 0;
41         }
42         if (nums[i]==','&&nums[i+1]=='\0'){
43             cout << "Please Enter Number Correctly!\nLike:\n1,2,3,4\nYou Have Something Like ', ,' Between In End Of Your Number\nReenter Your Number And Dont Enter ', ' At End\n";
44             system("pause");
45             return 0;
46         }
47     }
48     for(i=1;i<len;i++){//count number
49         if(nums[i]==','){
50             l++;
51         }
52     }
53     int num[l];
54     bool isNegative = false;
55     for (z=0,i=0;i<len;i++){//this cycle convert string to number and save number to N[] array
56     {
57         if (nums[i] == '-') {
58             isNegative = true;
59             continue;
60         }
61         if (nums[i] == ',') {
62             if (isNegative)
63                 j = -j;
64             num[z++] = j;
65             j = 0;
66             isNegative = false;
67             continue;
68         }
69         n = stoi(nums[i]);
70         j = j * 10 + n;
71     }
72     if (i == len - 1) {
73         if (isNegative)
74             j = -j;
75         num[z] = j;
76     }
77     }
78     for (int i = 0; i < l - 1; i++){//number sort!
79         for (int j = 0; j < l - i - 1; j++){
80             if (num[j] > num[j + 1]) {
81                 int temp = num[j];
82                 num[j] = num[j + 1];
83                 num[j + 1] = temp;
84             }
85         }
86     }
87     int f[1];
88     for(i=0;i<l;i++){// frequency calculate here
89         for(z=0,j=0;j<l;j++){
90             if(num[i]==num[j])
91                 z++;
92             f[i]=z;
93             z=0;
94         }
95     }
96     for (i=0;i<l;i++){//printer
97         if(num[i]==num[i+1])
98             continue;
99         cout << "F["<<num[i]<<"]="<<f[i]<<endl;
100    }
101    system("pause");
102    return 0;
103    }

```