ઍરે

'સી' પ્રોગ્રામિંગમાં અત્યાર સુધી આપણે મૂળભૂત ડેટા પ્રકાર જેવા કે int, float, char, double, વગેરેનો ઉપયોગ કર્યો. આ બધા ચલના પ્રકારો એક સમયે માત્ર એક જ કિંમત સાચવી શકે છે. એ કિંમત ઉપર પ્રક્રિયા કરવા માટે આપણને એક ચલની જરૂર પડે છે. તેથી જો આપણે પાંચ કિંમતો પર પ્રક્રિયા કરવી હોય તો એ માટે પાંચ જુદા જુદા ચલની જરૂર પડે. તો હવે એક એવા પ્રોગ્રામની કલ્પના કરો કે, જેમાં આપણને અનેક ડેટા પર પ્રક્રિયા કરવી પડતી હોય. આવી પરિસ્થિતિમાં, C ભાષા એક ખાસ પ્રકારની ઍરે ડેટા ટાઇપ આપે છે. આ પ્રકરણમાં આપણે એ શીખીશું કે C ભાષામાં ઉપલબ્ધ જુદા જુદા પ્રકારના ઍરેનો કેવી રીતે ઉપયોગ કરવો.

ઍરેની જરૂરિયાત (Need of arrays)

કલ્પના કરો કે, નીચે મુજબની જીવનની કેટલીક વાસ્તવિક પરિસ્થિતિ માટે આપણે C પ્રોગ્રામ લખવા છે :

- વર્ગના બધા વિદ્યાર્થીઓના ગુણ સાચવવા
- સુપર માર્કેટની બધી ચીજવસ્તુઓની કિંમત સાચવવી
- કર્મચારીઓ અને તેમના સંપર્ક સૂત્રની યાદી
- વર્ગના દરેક વિદ્યાર્થીની ઊંચાઈ અને વજનની માહિતી સાચવવી

દરેક વિદ્યાર્થીઓના ગુણની માહિતીનું વ્યવસ્થાપન કરવા marks0, marks1, marks2, marks3, marks4 .... અને marks59, એમ 60 વિદ્યાર્થીઓના ગુણ માટે 60 જુદા જુદા ચલ બનાવવાને બદલે આપણે એક જ ઍરે ચલ જેમ કે, marks[60] વ્યાખ્યાયિત કરવી શકાય. હવે આ ઍરેમાંથી અલગ અલગ વિદ્યાર્થીઓના ગુણ જાણવા માટે આપણે marks[0], marks[1], marks[2] ... એ પ્રમાણે ચલનો ઉપયોગ કરી શકીએ. કમ્પ્યૂટરની સ્મૃતિ(મેમરી)માં marks[60] નામનો ઍરે આકૃતિ 15.1માં દર્શાવ્યા મુજબ દેખાશે.

First element	Second element	Third element	....	....	Last element
marks[0]	marks[1]	marks[2]	marks[3]	.....	marks[59]

આકૃતિ 15.1 : સ્મૃતિમાં ઍરેના માળખાનો નમૂનો

ઍરેને સંબંધિત એવા નીચે મુજબના અગત્યના મુદ્દા નોંધો :

- તે એક જ ડેટાપ્રકારના ઘટકોનો સમૂહ છે.
- તે નિયત કદ ધરાવતા ઘટકોનો ક્રમિક સમૂહ છે. ઍરે સળંગ સ્મૃતિસ્થાનો રોકે છે.
- ઍરેનો કોઈ પણ ઘટક તેના અનુક્રમ દ્વારા ઉપયોગમાં લઈ શકાય છે. ઍરેના નામની પાછળ ચોરસ કૌંસમાં દર્શાવાતા અનુક્રમ(Index number)ને 'સબસ્ક્રિપ્ટ' (subscript) તરીકે પણ ઓળખવામાં આવે છે.
- આ સબસ્ક્રિપ્ટ તરીકે કોઈ પણ પૂર્ણાંક અથવા પૂર્ણાંક પદાવલિ જ હોવી જોઈએ.
- સબસ્ક્રિપ્ટનો ક્રમ શૂન્યથી શરૂ થાય છે.

ઍરેના પ્રકારો (Types of arrays)

'સી' ભાષામાં ઍરેને નીચે મુજબ બે પ્રકારોમાં વિભાજિત કરી શકાય :

(i) એક પરિમાણીય ઍરે (Single or One dimensional array)

(ii) બહુપરિમાણીય ઍરે (Multi dimensional array)

એક પરિમાણીય ઍરેમાં એક ઊભી હરોળ અથવા એક આડી હરોળ હોય છે જ્યારે બહુપરિમાણીય ઍરેમાં એક કે વધુ આડી હરોળ અથવા એક કે વધુ બહુ ઊભી હરોળ હોય છે. ઊભી અને આડી હરોળની સંખ્યા પ્રોગ્રામની જરૂરિયાત અનુસાર ઉપયોગકર્તા દ્વારા નક્કી કરવામાં આવે છે. આપણે માત્ર C ભાષા દ્વારા આપવામાં આવતા એક પરિમાણીય અને દ્વિપરિમાણીય ઍરેની જ ચર્ચા કરીશું.

એકપરિમાણીય ઍરે વ્યાખ્યાયિત કરવા (Declaration of single dimensional array)

C ભાષામાં ઍરેને વ્યાખ્યાયિત કરવા સ્મૃતિ ઘટકોમાં આપણે જે પ્રકારનો ડેટા સાચવવા માંગતા હોઈએ તે અને ઍરેમાં કેટલા ઘટકો રાખવા છે તેને ધ્યાને રાખી આપણે નીચે મુજબ ઍરે વ્યાખ્યાયિત કરીએ છીએ :

datatype arrayname[size];

- ***datatype:*** ઍરે ક્યા પ્રકારના ઘટકો સાચવે તે નક્કી કરે છે. જો આપણે ઍરેમાં શાબ્દિક માહિતી (characters) સાચવવા ઇચ્છતા હોઈએ તો ઍરેનો પ્રકાર 'char' હોય. એ જ રીતે, પૂર્ણાંક (integers) સંખ્યા સાચવવી હોય તો ઍરેનો પ્રકાર 'int' હોય. ટૂંકમાં, datatype તરીકે C ભાષાનો કોઈ પણ માન્ય ડેટાપ્રકાર હોઈ શકે.
- ***arrayname:*** આ પ્રોગ્રામર દ્વારા ઍરેને આપવામાં આવતું નામ છે. આ નામ તરીકે કોઈ પણ અક્ષરોનો સમૂહ ચાલી શકે. પરંતુ એવી સલાહ આપવામાં આવે છે કે, ઍરેને એક અર્થસભર નામ આપવું જોઈએ. ઍરેનું નામ બને ત્યાં સુધી પ્રોગ્રામ દ્વારા ઍરેમાં શું સાચવવામાં આવે છે તેના સંદર્ભમાં હોવું જોઈએ.
- ***size:*** size તરીકે આપવામાં આવતી કિંમત દ્વારા જ ઍરેમાં કેટલા ઘટકો સાચવી શકાશે તે નક્કી થાય છે. આ size તરીકે શૂન્યથી મોટો કોઈ પણ પૂર્ણાંક અચલ હોવો આવશ્યક છે.

ઉદાહરણ તરીકે, 60 વિદ્યાર્થીઓના ગુણની વિગતો સાચવવા આપણે નીચે મુજબનું વિધાન લખી શકીએ :

int marks[60];

અહીં એ નોંધ લો કે, વ્યાખ્યાયિત કરાયેલો marks નામનો ઍરે માત્ર 60 પૂર્ણાંક સંખ્યાઓ સંગ્રહવા માટે જ સક્ષમ છે. શાબ્દિક માહિતી, અપૂર્ણાંક સંખ્યાઓ અને વધુ મોટી કિંમતો (double) સાચવી શકે તેવા ઍરેના ઉદાહરણ નીચે વ્યાખ્યાયિત કરેલ છે :

char string [20];

float percentages [20];

double numbers [20];

એકપરિમાણીય ઍરેને કિંમતો આપવી (Assigning values to single dimensional arrays)

કોઈ પણ ઍરેને પ્રારંભિક કિંમતો આપવા માટે C ભાષામાં બે રીતો છે :

(i) કંપાઈલેશન સમયે ઍરેને પ્રારંભિક કિંમતો આપવી.

(ii) પ્રોગ્રામ ચાલે ત્યારે ઍરેને પ્રારંભિક કિંમતો આપવી.

કંપાઈલેશન સમયે ઍરેને પ્રારંભિક કિંમતો આપવી (Compile time array initialization)

ઍરેને વ્યાખ્યાયિત કરતી વખતે કોઈ સામાન્ય ચલની જેમ જ કિંમતો આપી શકાય છે. વિવિધ ઍરે ઘટકોને કિંમતો આપવા માટેની સામાન્ય વાક્યરચના નીચે મુજબ છે :

datatype arrayname[size] = {value1, value2, ... , valueN};

અહીં, *value1, value2, ... , valueN* એ ઍરેના એક પછી એક આગળના ઘટકો માટે પ્રારંભિક કિંમત પૂરી પાડે છે. નીચે કેટલાંક ઉદાહરણ આપ્યાં છે જેમાં આપણે ઍરેના તમામ ઘટકોનું કદ નક્કી કરી તેને કિંમતો આપી શકીએ છીએ.

int marks[5] = {60, 65, 70, 75, 80};

આ ઉદાહરણ marks નામનો ઍરે ચલ વ્યાખ્યાયિત કરશે, જે પાંચ ઘટકો સાચવી શકશે. આ ઘટકોની કિંમત

marks[0] = 60, marks[1] = 65, marks[2] = 70, marks[3] = 75 અને marks[4] = 80 ગોઠવાશે. આવી રીતે ઍરે વ્યાખ્યાયિત કરવાના કિસ્સામાં જો ઍરેના કદ કરતાં કિંમતોની યાદી નાની હશે, તો શરૂઆતના એક પછી એક ઘટકોને કિંમત આપવામાં આવશે અને પાછળના ઘટકોની કિંમત શૂન્ય રાખવામાં આવશે. ઉદાહરણ તરીકે,

```
int marks[5] = {60, 65, 70};
```

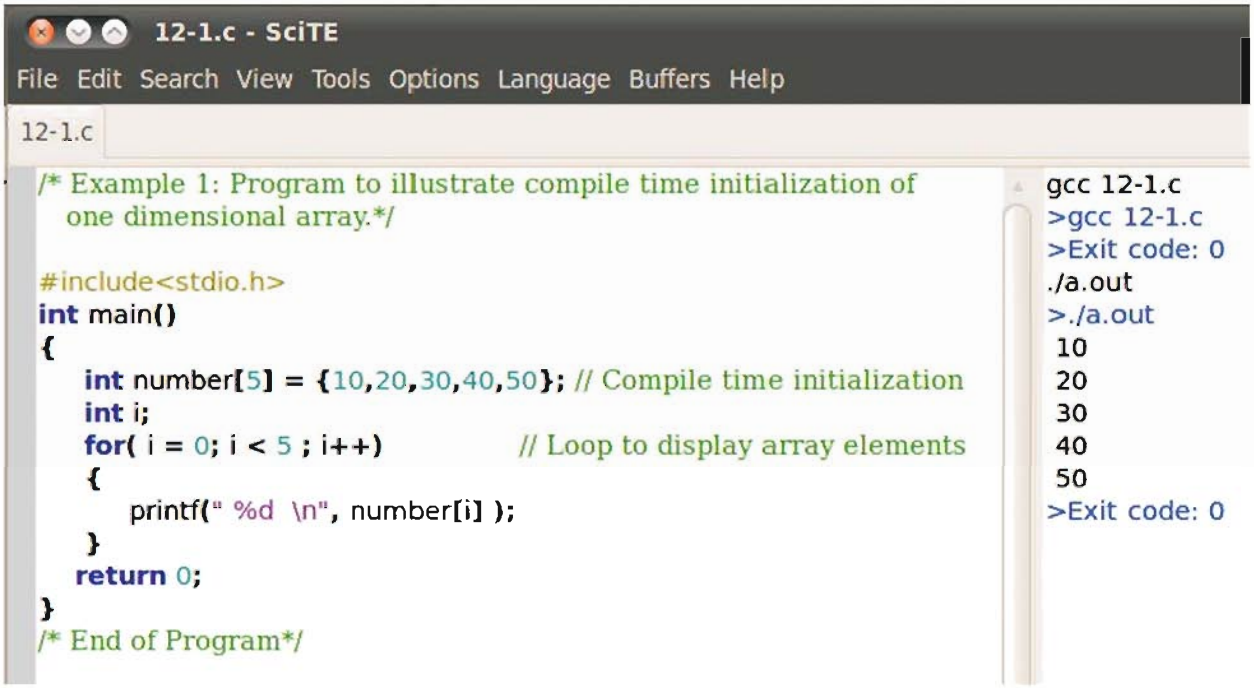
પ્રથમ ત્રણ ઍરે ઘટકોની પ્રારંભિક કિંમત marks[0] = 60, marks[1] = 65 અને marks[2] = 70 થશે. જ્યારે એ પછીના marks[3] અને marks[4] ઘટકની પ્રારંભિક કિંમત 0 અપાશે.

જો ઍરેને વ્યાખ્યાયિત કરતી વખતે જ પ્રારંભિક કિંમત આપવામાં આવે તો ઍરેનું કદ દર્શાવવું મરજિયાત છે. જો આપણે ઍરેનું કદ નહીં દર્શાવીએ તો કંપાઇલર આપમેળે છગડિયા કૌંસ({})ની અંદર આપેલી કિંમતોની સંખ્યા ગણીને નક્કી કરી લેશે. ઉદાહરણ તરીકે, નીચેનું વિધાન

```
int marks[ ] = {60, 65, 70, 75, 80};
```

પાંચ ઘટકો ધરાવતો marks નામનો ઍરે વ્યાખ્યાયિત કરશે જેને છગડિયા કૌંસમાં આપેલી કિંમતો દ્વારા પ્રારંભિક કિંમતો આપવામાં આવશે.

તો ચાલો, ઉદાહરણ 15.1 સાથે એક પરિમાણીય ઍરેને કંપાઇલેશન સમયે પ્રારંભિક કિંમત આપવાના ખ્યાલનો અભ્યાસ કરીએ. આકૃતિ 15.2 ઉદાહરણ 15.1નું કોડ-લિસ્ટિંગ અને પરિણામ રજૂ કરે છે.



આકૃતિ 15.2 : ઉદાહરણ 15.1નું કોડ-લિસ્ટિંગ અને પરિણામ

સમજૂતી

main વિધેય પછીની પ્રથમ લીટીમાં number નામનો ઍરે વ્યાખ્યાયિત કરવામાં આવ્યો છે અને કંપાઇલેશન સમયે જ તમામ ઍરે ઘટકોને છગડિયા કૌંસમાં આપેલી કિંમતો વડે પ્રારંભિક કિંમત આપવામાં આવે છે. for લૂપ number નામના ઍરેના અલગ-અલગ ઘટકને દર્શાવે છે.

અમલ દરમ્યાન ઍરેને પ્રારંભિક કિંમતો આપવી

જ્યારે આપણને પ્રોગ્રામનો અમલ કરતી વખતે ઉપયોગકર્તા પાસેથી કિંમતો મેળવવાની જરૂર પડતી હોય, તો આપણે અમલ દરમ્યાન ઍરેની પ્રારંભિક કિંમત આપવાની પદ્ધતિનો ઉપયોગ કરી શકીએ. ઍરેમાં સંગૃહિત કિંમતોનો કેવી રીતે ઉપયોગ કરવો તે પણ આ પ્રોગ્રામ દર્શાવે છે. આકૃતિ 15.3, ઉદાહરણ 15.2નું કોડ-લિસ્ટિંગ આપે છે, જ્યારે આકૃતિ 15.4 તેનું પરિણામ દર્શાવે છે.

```
12-2.c * SciTE
File  Edit  Search  View  Tools  Options  Language  Buffers  Help
12-2.c *

/* Example 2: Program to illustrate runtime array initialization and accessing values
   from one dimensional array. Program will read marks of five students and display the
   average marks.*/

#include<stdio.h>
int main()
{
        int marks[5], i, total=0;
        for( i = 0; i < 5; i++)                 // loop to read marks of five students
        {
                printf("Enter marks[%d] :", i);
                scanf("%d", &marks[i]);
        }
        for( i = 0; i < 5; i++)                 // loop to find total marks
        {
                total = total + marks[i];
        }
        printf("\nThe average marks of students are %d. \n", total/5);
        return 0;
}
/* End of Program*/
```

આકૃતિ 15.3 : ઉદાહરણ 15.2નું કોડ-લિસ્ટિંગ

```
kpp@ubuntu: ~
File  Edit  View  Terminal  Help
kpp@ubuntu:~$ gcc 12-2.c
kpp@ubuntu:~$ ./a.out
Enter marks[0] :5
Enter marks[1] :10
Enter marks[2] :15
Enter marks[3] :20
Enter marks[4] :25

The average marks of students are 15.
kpp@ubuntu:~$
```

આકૃતિ 15.4 : ઉદાહરણ 15.2નું પરિણામ

સમજૂતી

પ્રોગ્રામની પ્રથમ *for* લૂપ પાંચ વિદ્યાર્થીઓના ગુણ વાંચશે. બીજી *for* લૂપ marks નામના એરેના અલગ-અલગ ઘટક વાંચીને કુલ ગુણ ગણશે. છેલ્લું printf વિધેય વિદ્યાર્થીઓના સરેરાશ ગુણ દર્શાવશે.

કેરેક્ટર એરે તરીકે શબ્દમાળા

શબ્દમાળા એ અક્ષરોની હારમાળા છે, જે સામાન્ય રીતે એક ઘટક તરીકે લેવામાં આવે છે. C ભાષા ડેટાના એક પ્રકાર તરીકે શબ્દમાળા માટે string ડેટા પ્રકારને સમર્થન આપતી નથી. જો કે, તેમ છતાં તે આપણને કેરેક્ટર એરે

તરીકે કોઈ પણ શાબ્દિક માહિતીને રજૂ કરવાની છૂટ આપે છે. શાબ્દિક ચલ એ C ભાષાનું કોઈ પણ માન્ય ચલનું નામ છે, અને તે હંમેશાં કેરેક્ટર ઍરે તરીકે જ વ્યાખ્યાયિત કરવામાં આવે છે. શાબ્દિક ચલને વ્યાખ્યાયિત કરવા માટેનું સામાન્ય સ્વરૂપ નીચે મુજબ છે:

```
char stringname [size];
```

અહીં, size એ stringname નામના ચલમાં અક્ષરોની સંખ્યા દર્શાવે છે. નીચે કેટલાક માન્ય શાબ્દિક ચલને વ્યાખ્યાયિત કરવાનાં ઉદાહરણ આપેલ છે :

```
char student_name [20];
char city [50];
char state [20];
```

અહીં એ નોંધવું જરૂરી છે કે C ભાષામાં શાબ્દિક માહિતીનો અંત ખાલી અક્ષર (null character) તરીકે ઓળખાતા '\0'; ખાસ ચિહ્નથી આવે છે. આ ખાસ ચિહ્ન પ્રોગ્રામને શાબ્દિક માહિતીનો અંત ઓળખવામાં મદદરૂપ બને છે. આમ, કોઈ પણ શાબ્દિક માહિતીના અંતે '\0' (null character) ચિહ્ન મૂકવું જરૂરી હોવાથી, આપણી શાબ્દિક માહિતીના બધા શબ્દોને સમાવી શકાય તે માટે કેરેક્ટર ઍરેમાં શબ્દોની કુલ સંખ્યા ઉપરાંત એક વધારાના સંગ્રહસ્થાનની જરૂર પડે છે.

સાંખ્યિક ઍરેની જેમ જ, શાબ્દિક ઍરેને પણ કંપાઇલ કરતી વખતે અથવા પ્રોગ્રામના અમલ દરમ્યાન એમ બે રીતે પ્રારંભિક કિંમતો આપી શકાય છે. કેરેક્ટર ઍરેને નીચે મુજબ જુદી જુદી રીતે પ્રારંભિક કિંમતો આપી શકાય :

```
char student_name[6] = "PURVA";
char student_name[6] = {'P', 'U', 'R', 'V', 'A', '\0'};
```

student_name નામના ઍરે કદ 6 ઘટકોનું રાખવા પાછળનું કારણ એ છે કે, PURVA શબ્દમાં પાંચ અક્ષર હોવાથી તેના 5 અને છેલ્લે null character ('\0') મૂકવા એક વધારાનો ઘટક એમ કુલ મળી 6 ઘટક. અહીં એ નોંધવું જોઈએ કે, જ્યારે તેના ઘટકોની યાદી રજૂ કરીને આપણે કેરેક્ટર ઍરેને પ્રારંભિક કિંમતો આપીએ ત્યારે શાબ્દિક માહિતીના અંતે null character '\0' દર્શાવવું જરૂરી છે. આપણે કદ દર્શાવ્યા વગર પણ નીચે મુજબ કેરેક્ટર ઍરે વ્યાખ્યાયિત કરી શકીએ :

```
char state[ ] = {'G', 'U', 'J', 'A', 'R', 'A', 'T', '\0'};
```

આ કિસ્સામાં state નામનો કેરેક્ટર ઍરે વ્યાખ્યાયિત થઈ જશે અને તેને 8 પ્રારંભિક કિંમતો અપાઈ જશે. નીચેનાં વિધેયોનો ઉપયોગ કરીને આપણે અગાઉ ચર્ચા કરેલ કેરેક્ટર ઍરેની માહિતી જોઈ શકીએ :

```
printf("The name of student is %s", student_name);
printf("The name of state is %s", state);
printf("The first character of your name is %c", student_name[0]);
```

બહુપરિમાણીય ઍરે (Multi dimensional array)

એકપરિમાણીય ઍરે માહિતીને રૈખિક ક્રમમાં જાળવે છે. જો કે, કેટલીક ઉપયોગી પદ્ધતિઓ (જેવી કે, સાંખ્યિક છબી, ચેસ રમતનું બોર્ડ વગેરે) સાથે સંકળાયેલ માહિતી દ્વિપરિમાણીય હોય છે. પ્રોગ્રામમાં આ પ્રકારની પદ્ધતિનો વિચાર કરતી વખતે આપણને બહુપરિમાણીય ઍરે બનાવવાની છૂટ આપે છે. અગાઉ જણાવ્યા મુજબ, બહુપરિમાણીય ઍરેમાં એક કરતાં વધુ ઊભી અને આડી હરોળ હોય છે. C ભાષામાં આપણે દ્વિપરિમાણીય, ત્રિપરિમાણીય અથવા N-પરિમાણીય ઍરે બનાવી શકીએ. બહુપરિમાણીય ઍરેનું સૌથી સાદું સ્વરૂપ દ્વિપરિમાણીય ઍરે છે. દ્વિપરિમાણીય ઍરેમાં આડી હરોળ માટે એક અને ઊભી હરોળ માટે એક એમ બે સબસ્ક્રિપ્ટ હોય છે. આ પ્રકરણમાં આપણે માત્ર દ્વિપરિમાણીય ઍરેની જ ચર્ચા કરીશું.

દ્વિપરિમાણીય ઍરેને વ્યાખ્યાયિત કરવો

અત્યાર સુધી આપણે એકપરિમાણીય ઍરેની ચર્ચા કરી, જે તેની અંદર ઘણી કિંમતો સાચવી શકે. જીવનમાં એવી ઘણી પરિસ્થિતિ હોય છે કે જેમાં આપણને ડેટાનું કોષ્ટક સાચવવાની જરૂર પડે. કોષ્ટક 15.1માં આપેલ કિસ્સા અંગે વિચારો. તે સોમવારથી શુક્રવાર દરમિયાન જુદા જુદા ત્રણ સેલ્સમૅન દ્વારા વેચવામાં આવેલ વસ્તુની વિગત દર્શાવે છે.

	Monday	Tuesday	Wednesday	Thursday	Friday
Salesman1	100	150	200	250	200
Salesman2	200	250	300	350	300
Salesman3	150	200	250	300	250

કોષ્ટક 15.1 : વેચાણની વિગતો

આ કોષ્ટક જુદી જુદી 15 કિંમતો ધરાવે છે. અહીં એકપરિમાણીય ઍરે વ્યાખ્યાયિત કરવો સલાહભર્યો નથી. દ્વિપરિમાણીય ઍરેનો ઉપયોગ કરવો એ વધુ સારો વિકલ્પ છે. C ભાષાનો દ્વિપરિમાણીય ઍરે વ્યાખ્યાયિત કરી કોષ્ટક 15.1ને સરળતાથી રજૂ કરી શકાય. દ્વિપરિમાણીય ઍરે નીચે મુજબ વ્યાખ્યાયિત કરી શકાય :

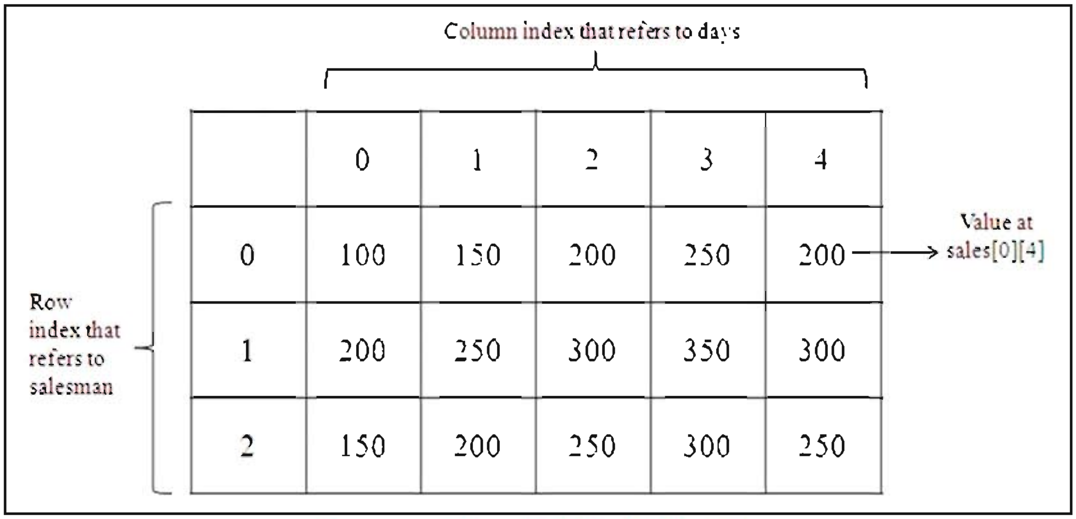
datatype arrayname [row size][column size];

row size અને *column size* શૂન્ય કરતાં મોટી પૂર્ણાંક અચલ સંખ્યા જ હોવી જોઈએ અને datatype તરીકે C ભાષાનો કોઈ પણ માન્ય ડેટા પ્રકાર હોઈ શકે. અહીં એ નોંધ કરો કે, C ભાષામાં બહુપરિમાણીય ઍરે એ આડી હાર અગ્રિમતા (row major) ધરાવે છે. બીજા શબ્દોમાં કહીએ તો, ઍરેને વ્યાખ્યાયિત કરતી વખતે દર્શાવવામાં આવતા બે કૌંસમાંથી પ્રથમ કૌંસ આડી હાર(rows)ની સંખ્યા નક્કી કરે છે.

કોષ્ટક 15.1માં દર્શાવેલ વિગતોને સાચવવા આપણે નીચે મુજબ દ્વિપરિમાણીય ઍરે વ્યાખ્યાયિત કરી શકીએ છીએ:

```
int sales[3][5];
```

અહીં, આડી હારની સંખ્યા 3 છે, જે સેલ્સમૅનને સંબોધે છે. (Salesman1 = 0, Salesman2 = 1 અને Salesman3 = 2) જ્યારે ઊભી હરોળની સંખ્યા 5 છે, જે અઠવાડિયાના જુદા જુદા વારને સંબોધવા ઉપયોગમાં લેવાશે. (Monday = 0, Tuesday = 1, Wednesday = 2, Thursday = 3 અને Friday = 4) આપણે એવું અનુમાન કર્યું છે કે, વેચાણના એકમોની સંખ્યા પૂર્ણાંક જ હશે. આકૃતિ 15.5 વેચાણની વિગતોની સ્મૃતિરચનાનો નમૂનો દર્શાવે છે.



આકૃતિ 15.5 : વેચાણની વિગતોની સ્મૃતિરચના

હવે, Salesman1 દ્વારા Mondayના રોજ કરવામાં આવેલ ચીજવસ્તુના વેચાણની વિગત ચકાસવા, આપણે ઍરે ઘટક sales[0][0]નો ઉપયોગ કરી શકીએ. sales[0][0] ખાતે સાચવેલ કિંમત 100 છે. એ જ રીતે, Mondayના રોજ Salesman2 અને Salesman3 દ્વારા કરવામાં આવેલ વેચાણની વિગત ચકાસવા આપણે અનુક્રમે ઍરે ઘટક sales[1][0] અને sales[2][0]નો ઉપયોગ કરી શકીએ. આકૃતિ 15.5માં દર્શાવ્યા મુજબ sales[0][4] સ્થાન પર સાચવેલી કિંમત 200 છે અને તે Fridayના રોજ Salesman1 દ્વારા કરવામાં આવેલ વસ્તુનું વેચાણ દર્શાવે છે.

દ્વિપરિમાણીય ઍરેને પ્રારંભિક કિંમતો આપવી

એકપરિમાણીય ઍરેની જેમ જ, દ્વિપરિમાણીય ઍરેને પણ આપણે કંપાઈલ કરતી વખતે અથવા પ્રોગ્રામના અમલ દરમિયાન પ્રારંભિક કિંમતો આપી શકીએ.

કંપાઈલ કરતી વખતે દ્વિપરિમાણીય ઍરેને પ્રારંભિક કિંમતો આપવી

કોષ્ટકના પ્રોગ્રામને કંપાઇલ કરતી વખતે કોષ્ટક 15.1 માં દર્શાવેલ ઍરેને પ્રારંભિક કિંમતો આપવાનું કાર્ય નીચે મુજબ કરી શકાય :

```
int sales[3][5]={
                {100, 150, 200, 250, 200},
                {200, 250, 300, 350, 300},
                {150, 200, 250, 300, 250}
        };
```

બહારના છગડિયા કૌંસની અંદર રહેલા છગડિયા કૌંસ, જે નિશ્ચિત સેલ્સમૅન માટેની આડી હરોળ દર્શાવે છે, તે મરજિયાત છે. નીચે મુજબ પ્રારંભિક કિંમતો આપવાની પદ્ધતિ અગાઉની પદ્ધતિ જેવું જ સમાન કાર્ય કરે છે.

```
int sales[3][5]={100, 150, 200, 250, 200, 200, 250, 300, 350, 300, 150, 200, 250, 300, 250};
```

પ્રોગ્રામના અમલ વખતે દ્વિપરિમાણીય ઍરેને પ્રારંભિક કિંમતો આપવી

પ્રોગ્રામનો અમલ કરતી વખતે કોષ્ટક 15.1માં દર્શાવેલ કોષ્ટકના ઍરેને પ્રારંભિક કિંમતો આપવાનું કાર્ય કોડ-લિસ્ટિંગ 15.1માં આપેલ પ્રોગ્રામ દ્વારા કરી શકાય.

```
int sales[3][5], row, column;
for( row = 0; row < 3; row++) {
        for (column = 0; column < 5; column++){
            printf("Enter sales item [%d][%d]:", row, column);
            scanf("%d", &sales[row][column]);
        }
}
```

કોડ-લિસ્ટિંગ 15.1 : અમલ દરમિયાન ડેટા ઉમેરવાનો પ્રોગ્રામ

આ પ્રોગ્રામ ત્રણ આડી હરોળ અને પાંચ ઊભી હરોળ સાથેનું sales નામનું દ્વિપરિમાણીય ઍરે વ્યાખ્યાયિત કરે છે. for લૂપ દ્વારા આપેલ તર્ક, sales નામના ઍરે માટે વિવિધ કુલ પંદર કિંમતો વાંચશે. sales ઍરેના અલગ-અલગ ઘટકોને કોડ-લિસ્ટિંગ 15.2 માં આપેલ પ્રોગ્રામ દ્વારા સ્ક્રીન પર દર્શાવી શકાય.

```
for( row = 0; row < 3; row++) {
    for (column = 0; column < 5; column++){
        printf("[%d][%d] : = %d \n", row, column, sales[row][column]);
    }
}
```

કોડ-લિસ્ટિંગ 15.2 : ઍરેની માહિતી દર્શાવવા માટેનો પ્રોગ્રામ

દ્વિપરિમાણીય ઍરેનું ઉદાહરણ

હવે આપણે પ્રોગ્રામના અમલ દરમિયાન એકપરિમાણીય અને દ્વિપરિમાણીય ઍરે વ્યાખ્યાયિત કરવા અને તેની કિંમતોનો ઉપયોગ દર્શાવતો પ્રોગ્રામ સમજીએ. આ પ્રોગ્રામ વિદ્યાર્થીઓના ગુણને સાચવવા માટે દ્વિપરિમાણીય ઍરે વ્યાખ્યાયિત કરશે. આ પ્રોગ્રામની સૂચનાઓ તેના પરિણામસહિત કોડ-લિસ્ટિંગ 15.3માં આપેલ છે.

```
/* Example 3 : Program to illustrate use of one અને two dimensional array. The program
will maintain marks of two students in three quizzes using two dimensional arrays. The
total marks of each student will be displayed.*/

#include<stdio.h>
#define STD 2    /*Number of rows for student data*/
#define QUIZ 3   /*Number of columns for quiz data*/

int main(){
        int marks[STD][QUIZ];              /* Marks array for 2 students અને 3 quizzes */
        int i, j, total_quiz[STD] = {0};

        for( i = 0; i < STD; i++)          /* Loop for student row */
        {
            for( j = 0; j < QUIZ;  j++)    /* Loop for quiz column */
                    {
              printf("Enter marks of student %d in quiz %d: ", i+1, j+1);
              scanf("%d", &marks[ i ][ j ]);          /*  Reading marks of quiz */
              total_quiz[ i] = total_quiz[ i] + marks[ i][ j];
            }
        }

          /* Logic to print student no., quiz marks અને total… */

        for ( i = 0 ; i < QUIZ; i++)       /* Loop for printing quiz number… */
        {
          printf(" \tQuiz %d" , i+1);
        }

        printf(" Total \n");
        for( i = 0; i < STD; i++)          /*  Loop for student row */
        {
          printf("Student %d:", i+1);
          for( j = 0; j < QUIZ;  j++)      /* Loop for quiz column */
          {
                  printf("%d \t", marks[ i ][ j ]);
           }
          printf("%d \n", total_quiz[ i]);
        }
        return 0;
}
/* End of Program*/
```

```
/*******************************************************************/
Output:

Enter marks of student 1 in quiz 1 : 20
Enter marks of student 1 in quiz 2 : 30
Enter marks of student 1 in quiz 3 : 40
Enter marks of student 2 in quiz 1 : 50
Enter marks of student 2 in quiz 2 : 40
Enter marks of student 2 in quiz 3 : 20

            Quiz1     Quiz2     Quiz3     Total

Student 1 :  20        30        40        90
Student 2 :  50        40        20        110
/*******************************************************************/
```

કોડ-લિસ્ટિંગ 15.3 : ઉદાહરણ 15.3ની સૂચનાઓ અને પરિણામ

સમજૂતી

અહીં, આપણે આપણા પ્રોગ્રામમાં બે પ્રતીકાત્મક અચલ STD અને QUIZનો ઉપયોગ કરેલ છે. પ્રોગ્રામ દ્વારા કેટલા વિદ્યાર્થીઓની સંખ્યાને સંભાળવાની છે તે STD દ્વારા નિયંત્રિત થાય છે. એ જ રીતે, પ્રોગ્રામ દ્વારા સંભાળવામાં આવનાર પ્રશ્નોની સંખ્યા QUIZ દ્વારા નક્કી થશે. આપણે, વિદ્યાર્થીઓના પ્રશ્નોના ગુણ સાચવવા marks નામનો દ્વિપરિમાણીય ઍરે વ્યાખ્યાયિત કર્યો છે. કોઈ એક વિદ્યાર્થી દ્વારા મેળવાયેલ તમામ પ્રશ્નોના કુલ ગુણ સાચવવા માટે total_quiz નામનો એકપરિમાણીય ઍરે વ્યાખ્યાયિત કરવામાં આવ્યો છે. total_quizને વ્યાખ્યાયિત કરતી વખતે જ તેના બધા ઘટકોને પ્રારંભિક કિંમત તરીકે શૂન્ય આપવામાં આવેલ છે. આપણી જરૂરિયાત અનુસાર for લૂપના વિધાનોને પુનરાવર્તિત કરવા માટે i અને j નામના બે ગણતરી દર્શક (counter) ચલ વ્યાખ્યાયિત કરવામાં આવ્યા છે.

વિદ્યાર્થીઓના પ્રશ્નોના ગુણ વાંચવા એક for લૂપની અંદર વળી બીજી for લૂપ(નેસ્ટેડ લૂપ)નો ઉપયોગ કરવામાં આવ્યો છે. લૂપની અંદર આપણે એક વિદ્યાર્થી દ્વારા તમામ પ્રશ્નોના કુલ ગુણની ગણતરી કરી છે. એ પછીની for લૂપનો ઉપયોગ પરિણામને સ્ક્રીન પર યોગ્ય રીતે રજૂ કરવા માટે કરવામાં આવ્યો છે. આપેલ પ્રોગ્રામ માત્ર બે વિદ્યાર્થીઓના ત્રણ પ્રશ્નોના ગુણને સાચવવા માટે જ સક્ષમ છે. પરંતુ, પ્રોગ્રામમાં STD અને QUIZની કિંમતો બદલીને આપણે આપણા પ્રોગ્રામને વધુ ક્રિયાશીલ બનાવી શકીએ.

ઍરેની સીમાની ચકાસણી

ઍરે એ C ભાષાની એક સક્ષમતા છે, પરંતુ અહીં એ નોંધવું અગત્યનું છે કે, C ભાષામાં ઍરેની સીમાની ચકાસણીની કોઈ વ્યવસ્થા નથી. ઍરેની સીમાની ચકાસણી એટલે કે પ્રોગ્રામમાં વ્યાખ્યાયિત કરેલ ઍરેની સીમાની ચકાસણી કરવી. ઍરે વ્યાખ્યાયિત કરતું નીચે આપેલું વિધાન ચકાસો.

```
int number[5];
```

આ વિધાન પાંચ પૂર્ણાંક કિંમતો સાચવતા number નામના ઍરેને વ્યાખ્યાયિત કરશે. આપણે number[0], number[1] ..... number[4]નો ઉપયોગ કરીને અલગ-અલગ ઍરે ઘટકોની કિંમતોનો ઉપયોગ કરી શકીશું. C ભાષા, ઍરેના [ ]માં કોઈ પણ ક્રમસંખ્યા લખવા માટે પ્રોગ્રામરને સત્તા આપે છે. એનો મતલબ, આપણે number [-1] અને number [6] અથવા અન્ય કોઈ પણ અમાન્ય અનુક્રમનો ઉપયોગ કરી શકીએ. ઍરેના અનુક્રમની સીમા ચકાસવા માટેની કોઈ સુવિધા ન હોવાને કારણે આપણને પ્રોગ્રામમાં કોઈ ભૂલ દર્શાવશે નહીં. પરંતુ, કદાચ પ્રોગ્રામ આપણને કિંમતરૂપે કોઈ ખોટી કિંમત આપશે અથવા પ્રોગ્રામ અધવચ્ચેથી બંધ (crash) થઈ જશે.

સારાંશ

આ પ્રકરણમાં આપણે C ભાષામાં એકપરિમાણીય અને દ્વિપરિમાણીય ઍરેના ઉપયોગ સંબંધિત અગત્યની લાક્ષણિકતાઓ શીખ્યા. આપણે એ યાદ કરવું જોઈએ કે, ઍરેનો ડેટા પ્રકાર C ભાષાનો કોઈ પણ માન્ય ડેટા પ્રકાર હોઈ શકે છે. ઍરેને વ્યાખ્યાયિત કરવામાં સબસ્ક્રિપ્ટ તરીકે કોઈ પણ પૂર્ણાંક કે પૂર્ણાંક પદાવલિ જ હોવી જરૂરી છે. ઍરેનો સબસ્ક્રીપ્ટ ક્રમ (index) શૂન્યથી શરૂ થાય છે. કેરેક્ટર ઍરેનો અંત ખાસ ચિહ્ન (null character) '\0' થી જ આવે છે. ઍરેની સીમાની ચકાસણીની કોઈ વ્યવસ્થા C ભાષામાં નથી. પ્રોગ્રામમાં માન્ય સબસ્ક્રિપ્ટ ક્રમનો ઉપયોગ કરવાની જવાબદારી પ્રોગ્રામરની છે.

સ્વાધ્યાય

1. પ્રોગ્રામમાં એકપરિમાણીય ઍરેનો ઉપયોગ કરવાના ફાયદા જણાવો.

2. એકપરિમાણીય અને દ્વિપરિમાણીય ઍરે વચ્ચેનો તફાવત સ્પષ્ટ કરો.

3. કંપાઇલ સમયે ઍરેના ઘટકોને પ્રારંભિક કિંમત આપવી અને પ્રોગ્રામના અમલ દરમિયાન ઍરેના ઘટકોને પ્રારંભિક કિંમત આપવી એટલે શું?

4. નીચેના પ્રોગ્રામ કોડમાં ભૂલો શોધો, જો ભૂલ જણાય તો તેને સુધારો અને તેનું પરિણામ જણાવો :

(a)
```
#include <stdio.h>
int main( ) {
    int num[3][3] = {{1, 2}, {3, 4}};
    printf("%d \n", number[1][1]);
    return 0;
}
```

(b)
```
#include<stdio.h>
void main()
{
    char str[ ] = 'INDIA'
    printf("%c %c %c %c  c", str[0], str[1], str[2], str[0], str[4]);
}
```

5. ખાલી જગ્યા પૂરો :

(a) દ્વિપરિમાણીય ઍરેને આડી હરોળ અને ____________ હોય છે.

(b) ઍરે ઘટક ____________ સ્મૃતિસ્થાનો રોકે છે.

(c) ઍરેની પાછળ ચોરસ કૌંસમાં લખવામાં આવતા અનુક્રમ (Index)ને ____________ તરીકે પણ ઓળખવામાં આવે છે.

(d) સબસ્ક્રિપ્ટ પૂર્ણાંક હોવો જોઈએ અથવા ____________ પદાવલિ હોઈ શકે.

(e) C ભાષાના કેરેક્ટર ઍરેમાં શાબ્દિક માહિતીનો ખાસ ચિહ્ન ____________ દ્વારા અંત લાવવામાં આવે છે.

6. નીચેનાં વિધાનો ખરાં છે કે ખોટાં તે જણાવો :

(a) ઍરેના ઘટકોને પ્રારંભિક કિંમતો માત્ર પ્રોગ્રામનો અમલ કરતી વખતે જ આપી શકાય છે.

(b) C ભાષામાં ઍરેના સબસ્ક્રિપ્ટની કિંમત 1થી શરૂ થાય છે.

(c) બહુપરિમાણીય ઍરેનું સાદું સ્વરૂપ એકપરિમાણીય ઍરે છે.

(d) C ભાષા string ડેટા ટાઇપનું સમર્થન કરે છે.

(e) દ્વિપરિમાણીય ઍરેને વ્યાખ્યાયિત કરતી વખતે પ્રથમ કૌંસ કુલ આડી હારની સંખ્યાનો નિર્દેશ કરે છે.

(f) int temperature[6] = {35, 32, 42}; એ ઍરેને પ્રારંભિક કિંમતો આપવાની માન્ય પદ્ધતિ છે.

(g) int num[ ] = {{1, 1}, {2, 2}}; એ પ્રારંભિક કિંમતો આપવાની માન્ય પદ્ધતિ છે.

7. નીચેના પ્રશ્નોના આપેલા વિકલ્પોમાંથી યોગ્ય વિકલ્પ પસંદ કરી જવાબ આપો :

(1) C ભાષામાં જો કોઈ ઍરેના કુલ ઘટકની સંખ્યા કરતાં મોટી સંખ્યાના સબસ્ક્રિપ્ટ દ્વારા માહિતી મેળવવાનો પ્રયત્ન કરવામાં આવે તો શું થશે?

(a) ઘટકને શૂન્ય બનાવી દેવાશે.

(b) કંપાઇલર ભૂલ દર્શાવશે.

(c) પ્રોગ્રામ અધવચ્ચેથી અટકી પડે અથવા ખોટી કિંમત આપશે.

(d) ઍરેનું કદ આપમેળે વધારી આપવામાં આવશે.

(2) નીચેના પ્રોગ્રામનું પરિણામ શું આવશે?

```
int num[5] = {1, 2, 3, 4, 5};
int i, j;
i = num[1];
j = num[2];
printf("%d, %d, %d", i, j, num[0]);
```

(a) 1, 2, 3 (b) 2, 3, 1 (c) 1, 2, 0 (d) 3, 4, 5

(3) નીચેના પ્રોગ્રામનું પરિણામ શું આવશે?

```
int num[5] = {1, 2, 3};
printf("%d, %d", num[0], num[3]);
```

(a) 1, 2 (b) 1, 0 (c) 1, 3 (d) 2, 3

પ્રાયોગિક સ્વાધ્યાય

નીચેનાં કાર્યો પાર પાડવા માટેનો C પ્રોગ્રામ લખો :

1. ઉપયોગકર્તા પાસેથી 5 × 5નાં બે કોષ્ટક વાંચો. આ બે કોષ્ટકોનો સરવાળો કરી ત્રીજું કોષ્ટક બનાવી તેનું પરિણામ સ્ક્રીન પર દર્શાવો.

2. નીચે દર્શાવ્યા મુજબ પ્રોગ્રામમાં ઍરેનો ઉપયોગ કરી 20 વિદ્યાર્થીઓના ગુણ વાંચો :

Roll No.	Marks
1	60
2	70
...	....
20	75

- બધા પૈકી સૌથી વધુ ગુણ ધરાવતા વિદ્યાર્થીનો રોલ નંબર દર્શાવો.
- 50 કરતાં વધુ ગુણ ધરાવતા કુલ વિદ્યાર્થીઓની સંખ્યા દર્શાવો.
- 50 કરતાં ઓછા ગુણ ધરાવતા કુલ વિદ્યાર્થીઓની સંખ્યા દર્શાવો.

3. એકપરિમાણીય ઍરેમાં 10 પૂર્ણાંક સંખ્યા વાંચો. ઍરેના ઘટકોને અવળા ક્રમમાં દર્શાવો. ઉદાહરણ તરીકે, જો ઉપયોગકર્તાનું ઈનપુટ 10, 20, 30, 40 હોય તો પરિણામ 40, 30, 20, 10 એમ આવવું જોઈએ.

4. ઍરેના N ઘટકોમાંથી મહત્તમ (maximum), લઘુતમ (minimum) અને સરેરાશ શોધો.

5. ઉપયોગકર્તા પાસેથી 3 × 3ના કોષ્ટકની પૂર્ણાંક સંખ્યા વાંચો. કોષ્ટકના દરેક ઘટકનો સરવાળો ગણીને દર્શાવો.