PROBLEM APPROACH

NILEENA K B

IV ECE

Dr.N.G.P Institute of Technology

CONTENTS

- PROBLEM STATEMENT
- FLOWCHART
- ALGORITHM
- CASE BLOCKS WITH EXPLANATION
- CONCLUSION

PROBLEM STATEMENT

Write a C++ program to create a list of software application details. The details of application include name, author, version, publishing year , price

Perform the following with respect to the list of application created.

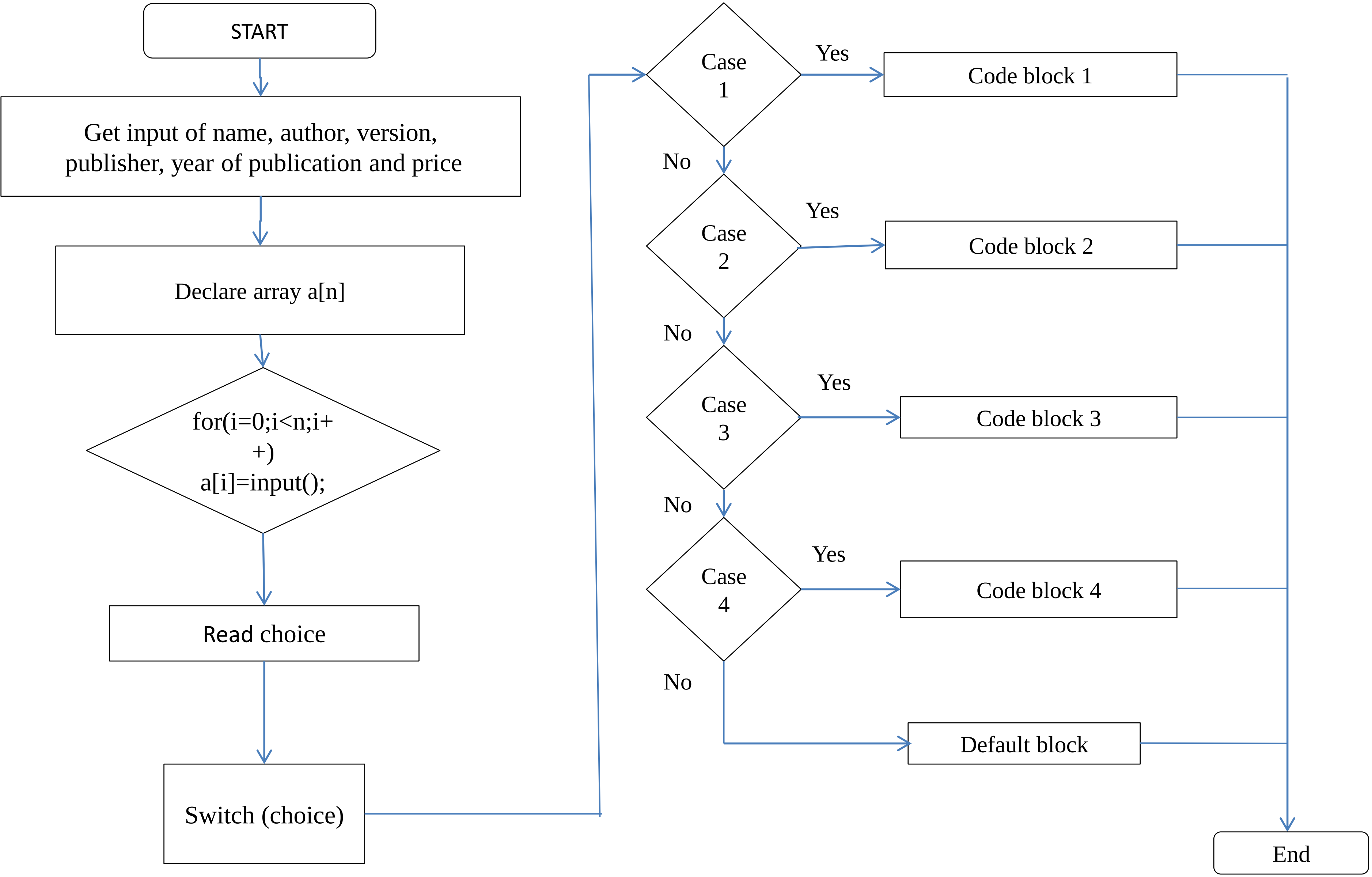
a) Display all the details of application by a given author.

b) Sort the details of application in the increasing order of price.

c) Display the details of applications published by a given publisher in a given year.

d) Sort the list of applications in the increasing order of two fields , author and publishing year of the books.

FLOWCHART


START
Get input of name, author, version, publisher, year of publication and price
Declare array a[n]
for(i=0;i<n;i+
+)
a[i]=input();
Read choice
Switch (choice)
Case
1
Yes
Code block 1
No
Case
2
Yes
Code block 2
No
Case
3
Yes
Code block 3
No
Case
4
Yes
Code block 4
No
Default block
End

ALGORITHM

- Start by creating a structure “app”.
- Initialize all the required details as given.(name, author, version, publisher, year of publishing, price)
- Print the details after getting the input from the user.
- Initialize n,i,j,y and s for the case blocks.
- Enter the number of software applications required. Get input from the user.
- Create an array of applications a[n].
- Enter the choices required.
- Using switch-case statements, case blocks are created.
  - Case 1: Print all applications of an author
  - Case 2 : Print the applications in increasing order of price.
  - Case 3 : To find all applications published by a publisher in a year
  - Case 4: To sort details of applications in increasing order of author and publishing year.

CASE BLOCKS

CASE 1:

-
  ```
  cout << "Enter the author : " << endl;
  cin >> s;
  cout << “Applications by " << s << " are as follows :" << endl;
  for(i=0;i<n;i++)
  {
  if(a[i].author==s)
  print(a[i]);
  }
  ```

EXPLANATION:

- The condition checks and compares the given input with the author of the software application.
- If it matches, the details of all the applications written by that author will be displayed.

CASE BLOCKS

CASE 2: (using swapping)

-
  ```
  for(i=0;i<n-1;i++)
  {
  for(j=0;j<n-i-1;j++)
  {
  if (a[j].price>a[j+1].price)
  {
  struct app temp;
  temp=a[j];
  a[j]=a[j+1];
  a[j+1]=temp;
  }
  }
  }
  ```

EXPLANATION

- Every price is compared to the previous one in the array.
- Temporary variable is used here for swapping operation.
- The details of applications are displayed starting with the application with the lowest price.

CASE BLOCKS

CASE 3:

```
cout << "Enter the publisher and year of
publishing : " << endl;
cin >> s;
cin >> y;
cout << “Applications published by " <<
s
<< " in the year " << y << " are as
follows :" << endl;
for(i=0;i<n;i++)
{
if(a[i].publisher==s && a[i].yearpub==y)
print(a[i]);
}
```

EXPLANATION:

- Declare two variables s and y for getting input from the user.
- The block checks for the given publisher and year of publishing.
- The details are displayed once it both matches.

CASE BLOCKS

CASE 4:

```
for(i=0;i<n-1;i++)
{
for(j=0;j<n-i-1;j++)
{
if (a[j].author>a[j+1].author)
{
struct app temp;
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
}
}
for(i=0;i<n;i++)
{
print(a[i]);
}
for(i=0;i<n-1;i++)
{
for(j=0;j<n-i-1;j++)
{
if (a[j].yearpub>a[j+1].yearpub)
{
struct app temp;
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
}
}
cout << “Applications sorted by year of
publishing are as follows :" << endl;
for(i=0;i<n;i++)
{
print(a[i]);
}
```

Swap function used in the previous block is used here for two fields (author and year of publication). They are printed in increasing order of the same.

CONCLUSION

Here, a structure with class ‘app’ is created along with a subclass ‘b’. With the help of array and switch-case statements, the given problem is solved and the list of application is displayed as desired by the user.

THANK YOU