Title: E-MARKETPLACE NITK

Team number: 30

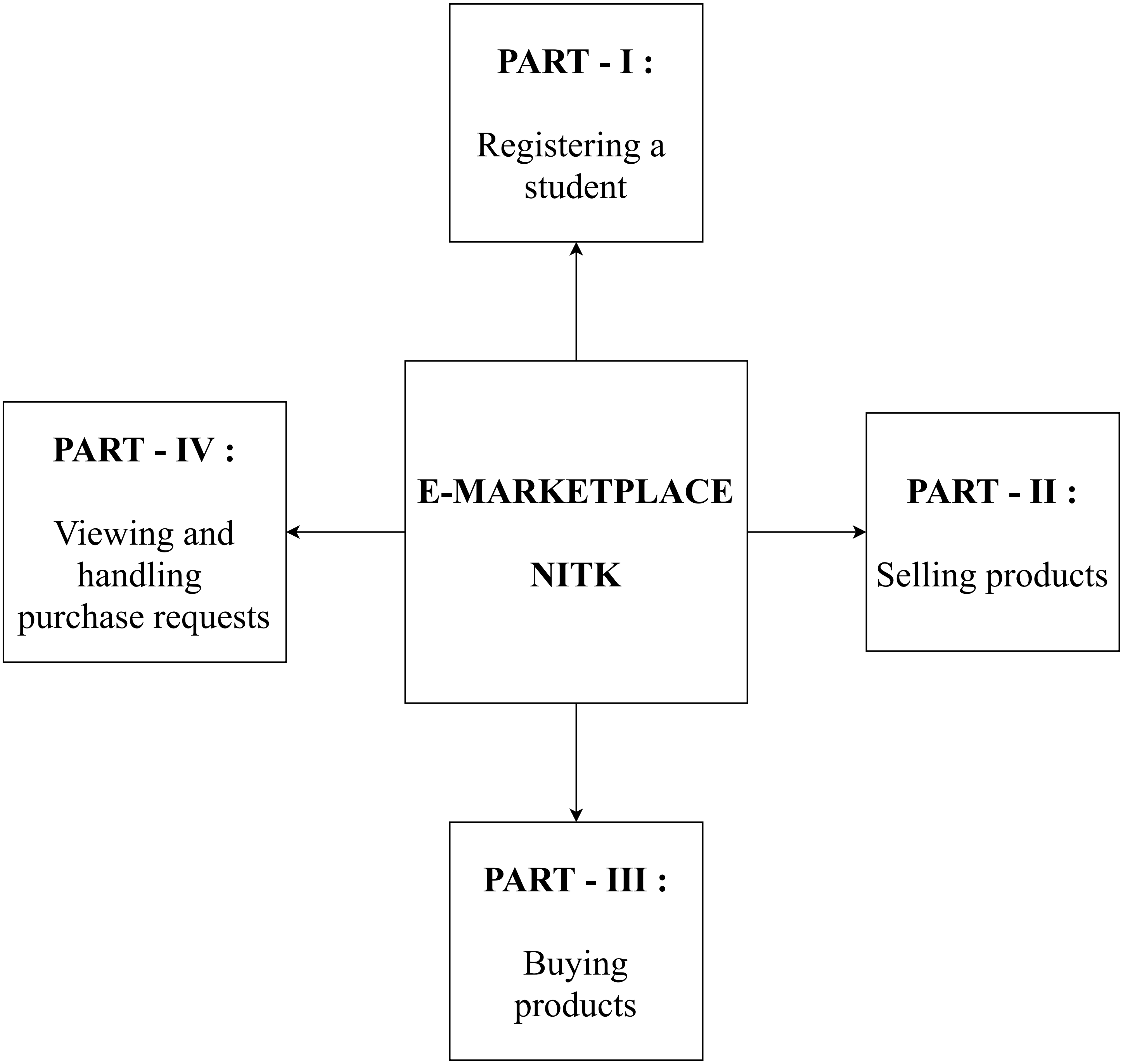
Team members:

1. R RAGHAVENDRA,
ROLL NUMBER: CH036,
PHONE: 9880234074,
EMAIL ID: ragavindrar@gmail.com

2. DHRUV KUMAR JHA,
ROLL NUMBER: ME128,
PHONE: 8073503626,
EMAIL ID: jhadhruv206@gmail.com

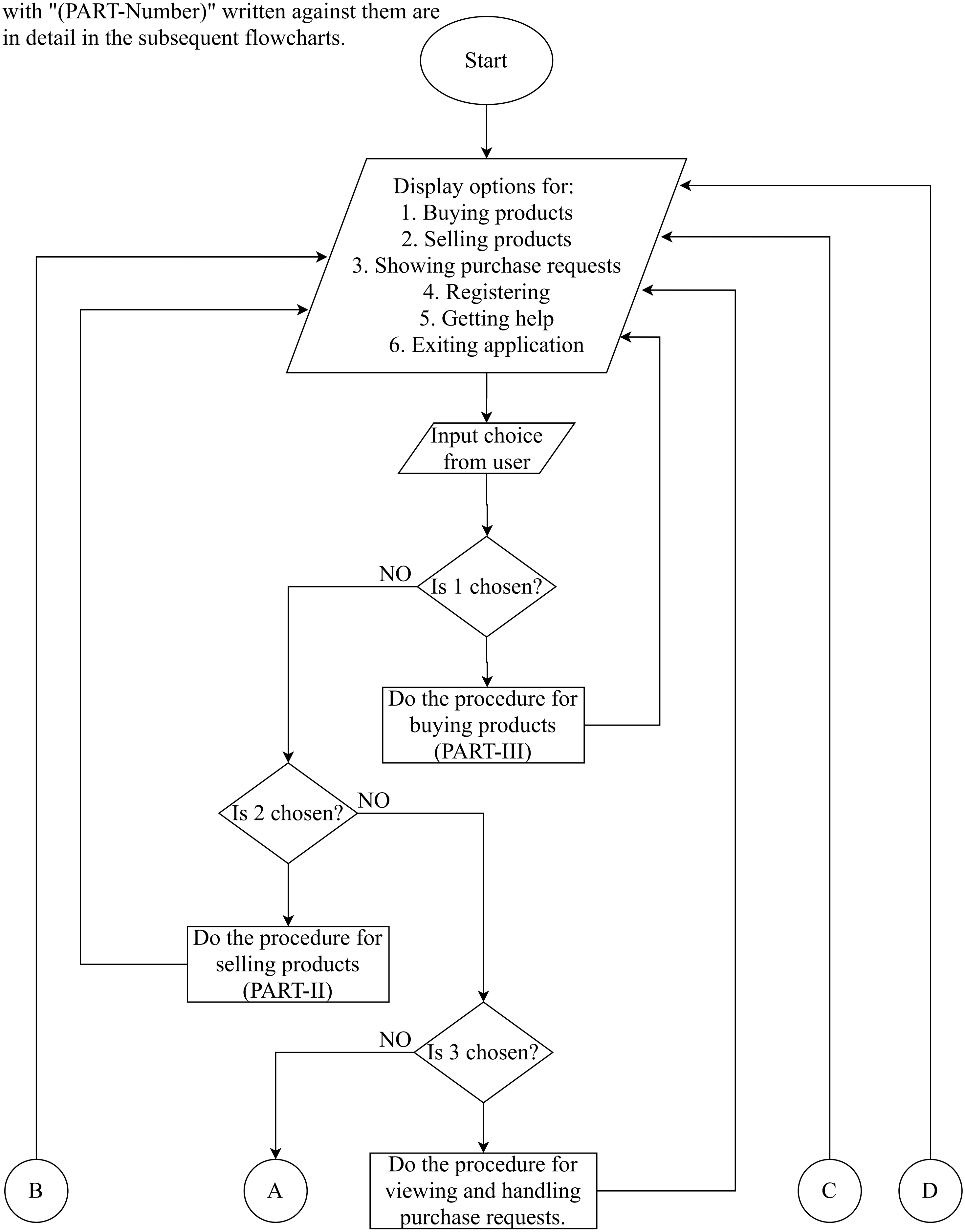
NOTE: This is only an overview of the project, identifying the major modules/ parts involved. Flowcharts for the overall implementation and each module/part are drawn from the next page onwards.

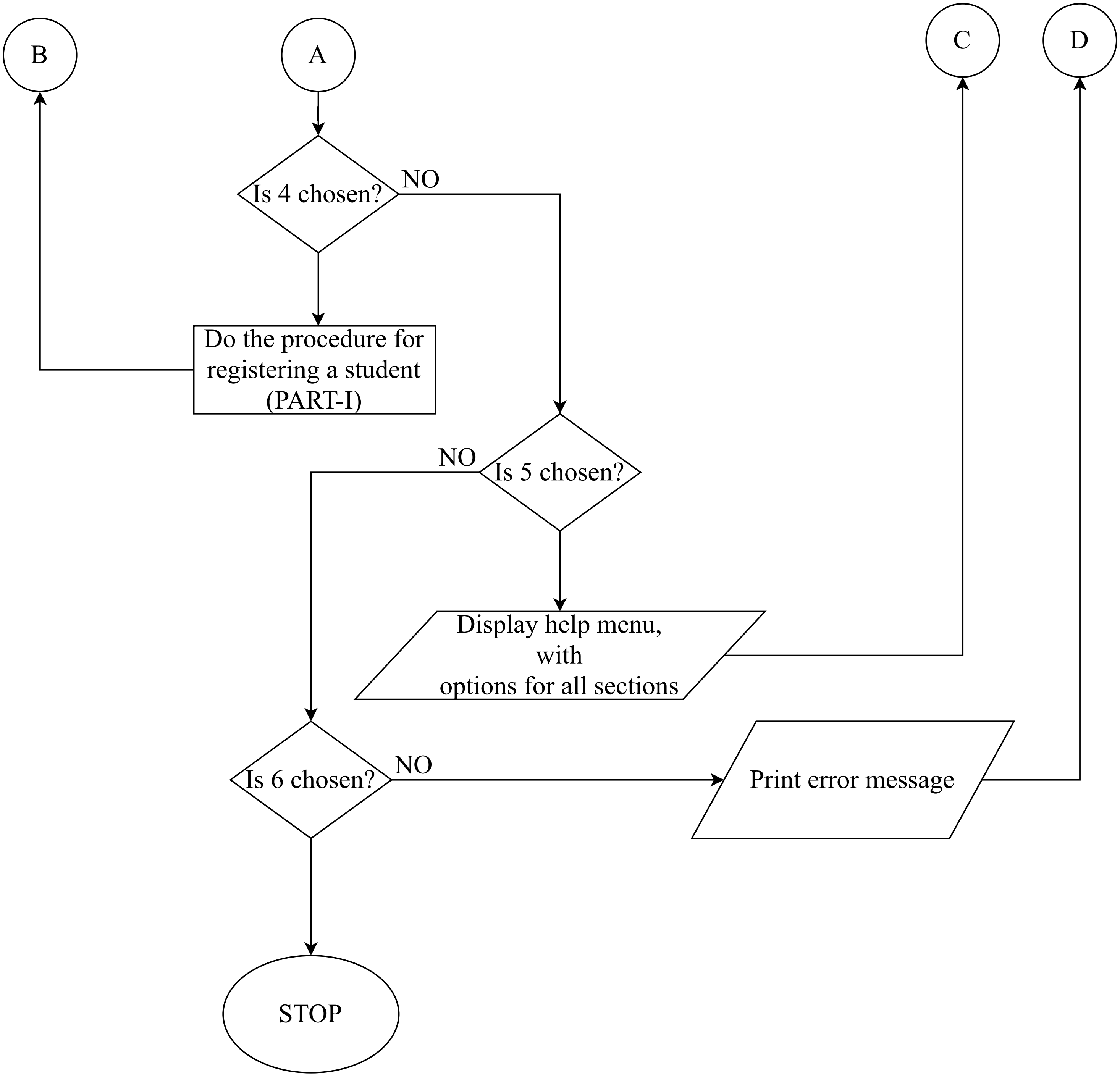
OVERVIEW OF THE PROJECT:



Overall implementation of the project:

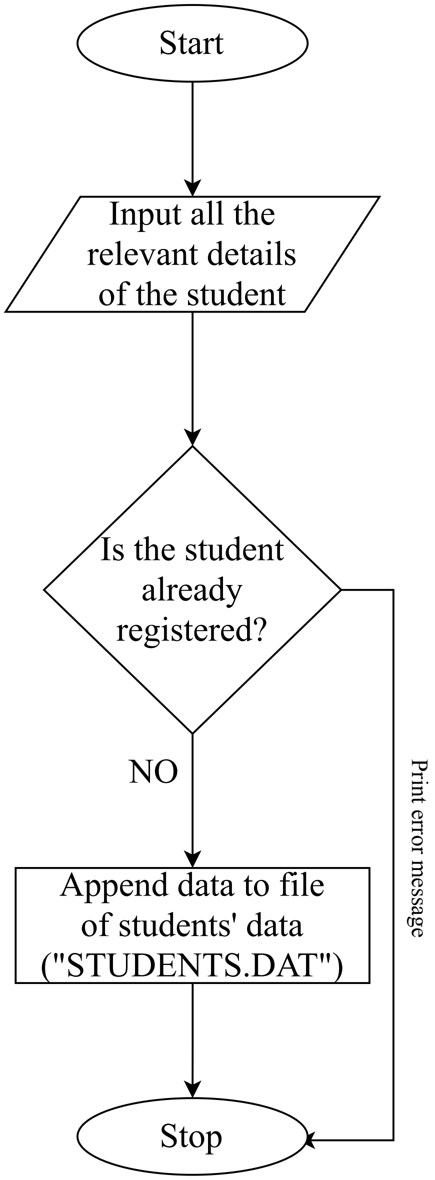
This flowchart is only to show the overall implementation of the project, that is, how the various modules are used by a user upon opening the application. The operations in this flowchart with "(PART-Number)" written against them are described in detail in the subsequent flowcharts.


Start
Display options for:
1. Buying products
2. Selling products
3. Showing purchase requests
4. Registering
5. Getting help
6. Exiting application
Input choice from user
Is 1 chosen?
NO
Do the procedure for buying products (PART-III)
Is 2 chosen?
NO
Do the procedure for selling products (PART-II)
Is 3 chosen?
NO
Do the procedure for viewing and handling purchase requests.
B
A
C
D

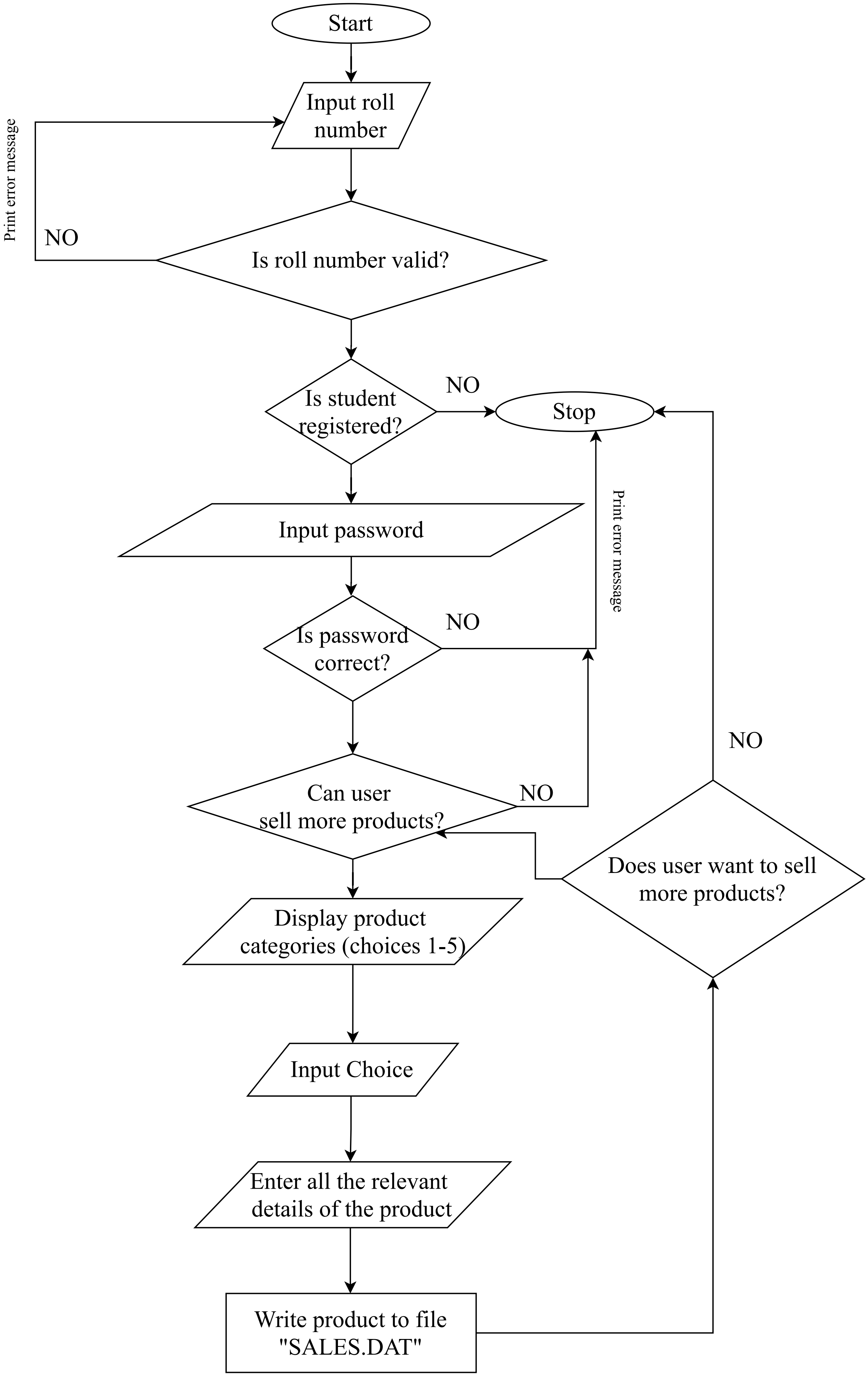

B
A
C
D
Is 4 chosen?
NO
Do the procedure for registering a student (PART-I)
NO
Is 5 chosen?
Display help menu, with options for all sections
Is 6 chosen?
NO
Print error message
STOP

PART-I:Procedure for registering a student.

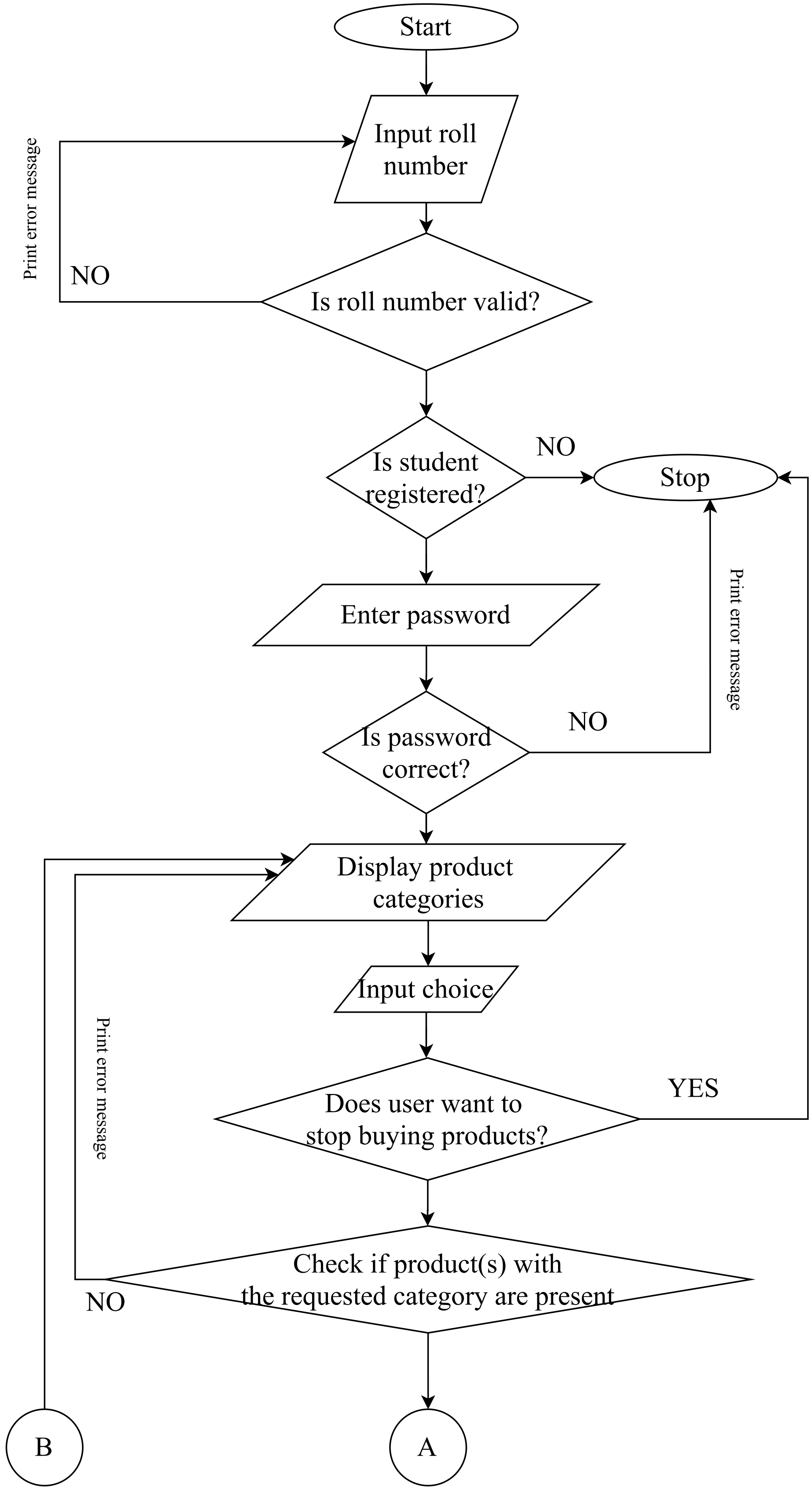
NOTE: This flowchart describes only the overall process of registering a student. The flowcharts for the functions used here are shown after the flowcharts of PARTS-II, III, IV of the project. The same applies to the flowcharts of PARTS-II, III, IV.

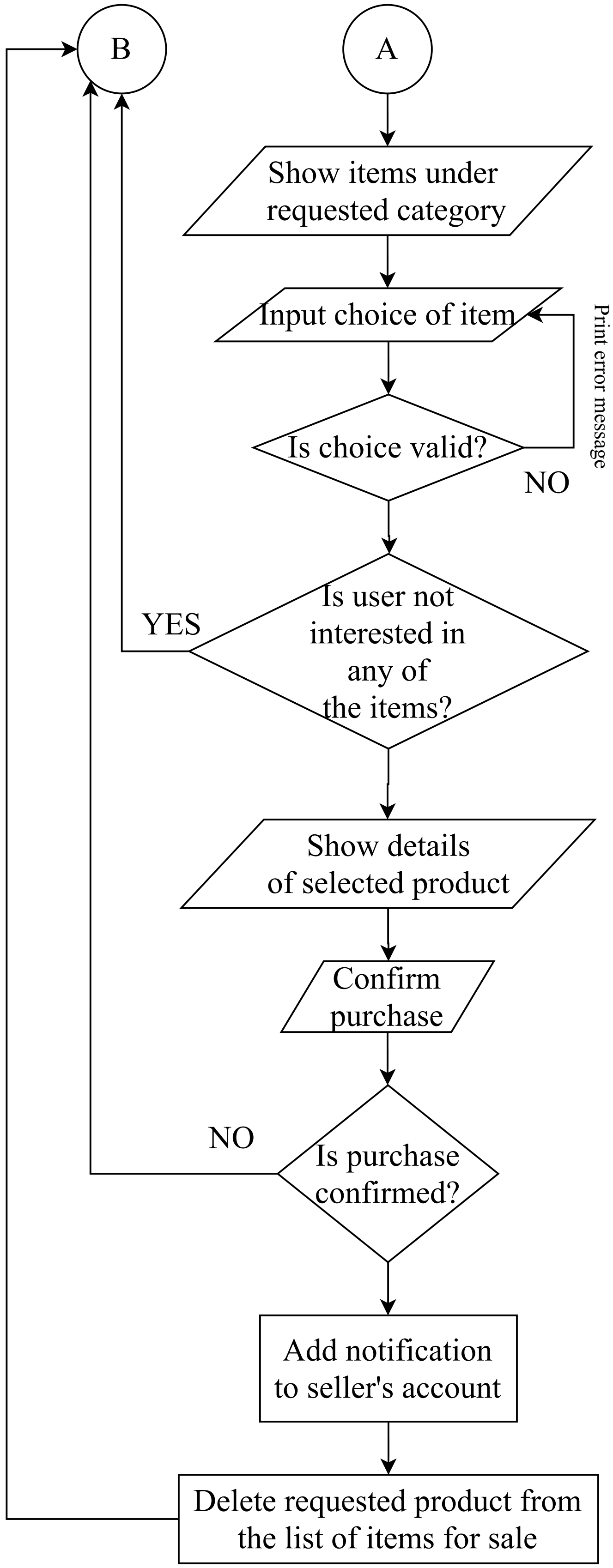

Start
Input all the relevant details of the student
Is the student already registered?
NO
Print error message
Append data to file of students' data ("STUDENTS.DAT")
Stop

PART-II:Procedure for selling products

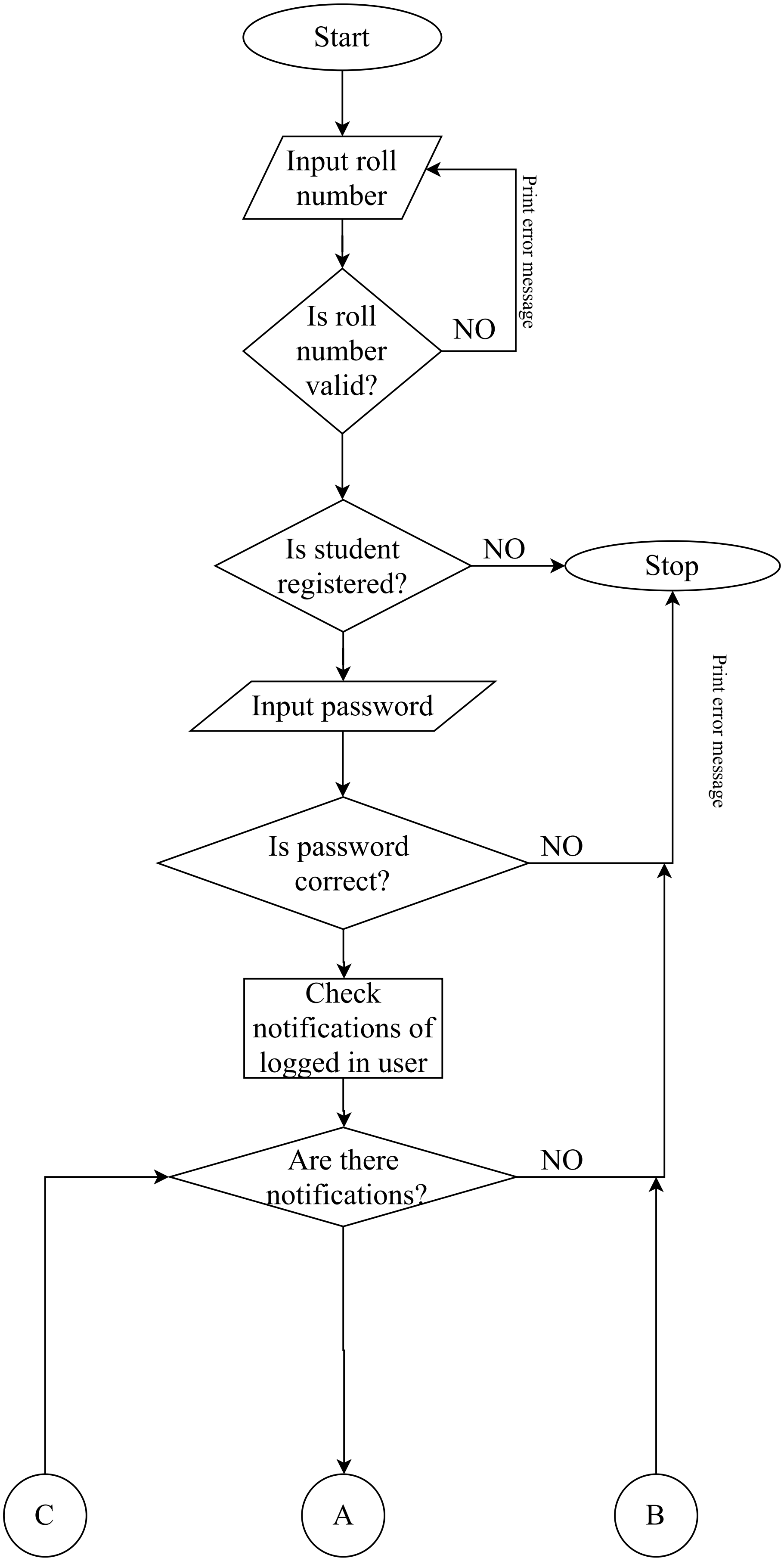

Start
Input roll number
Print error message
NO
Is roll number valid?
Is student registered?
NO
Stop
Input password
Print error message
Is password correct?
NO
Can user sell more products?
NO
NO
Does user want to sell more products?
Display product categories (choices 1-5)
Input Choice
Enter all the relevant details of the product
Write product to file "SALES.DAT"

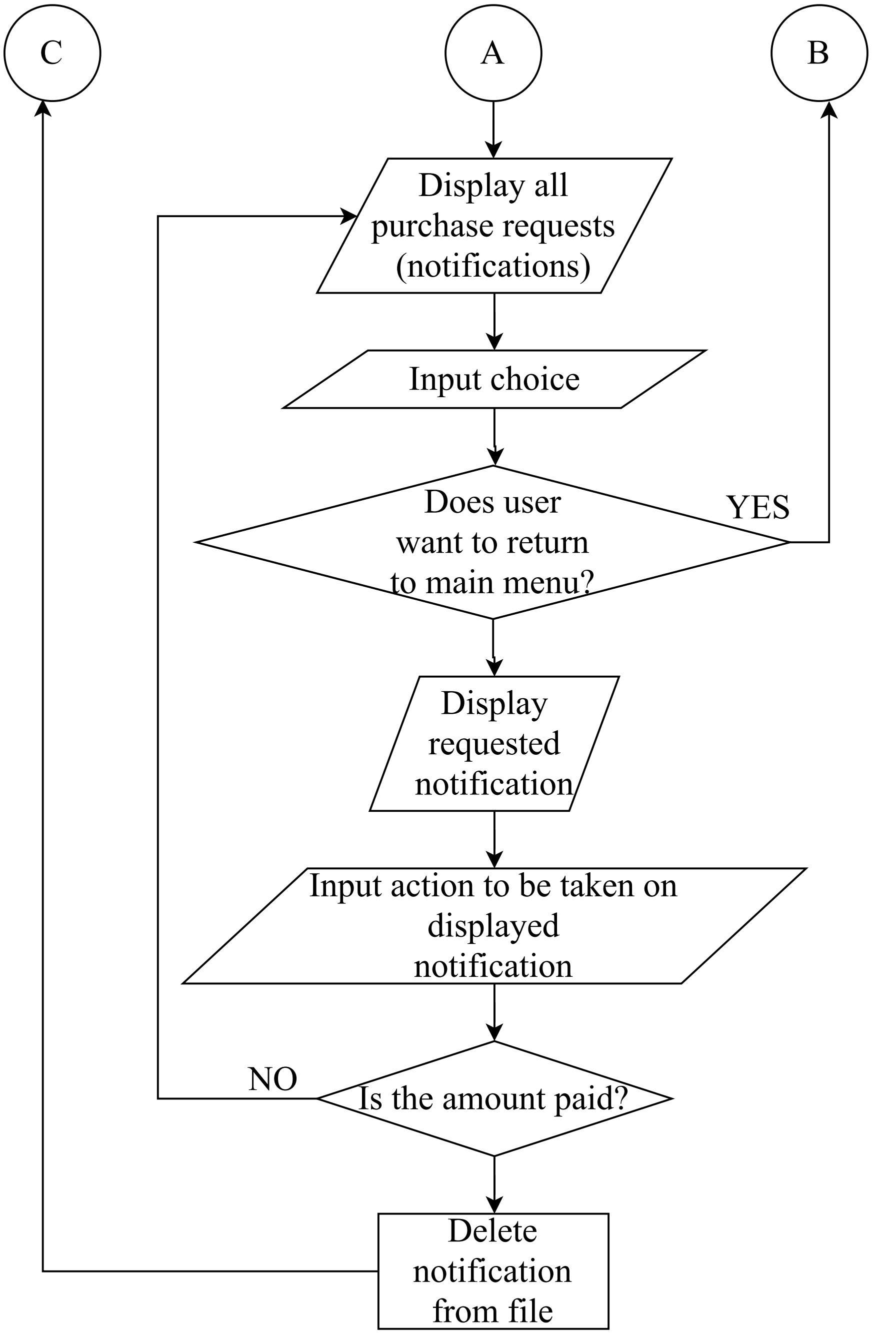
Part-III:Procedure for buying products




B
A
Show items under requested category
Input choice of item
Print error message
Is choice valid?
NO
YES
Is user not interested in any of the items?
Show details of selected product
Confirm purchase
NO
Is purchase confirmed?
Add notification to seller's account
Delete requested product from the list of items for sale

PART-IV:Procedure to view and handle purchase requests(notifications)



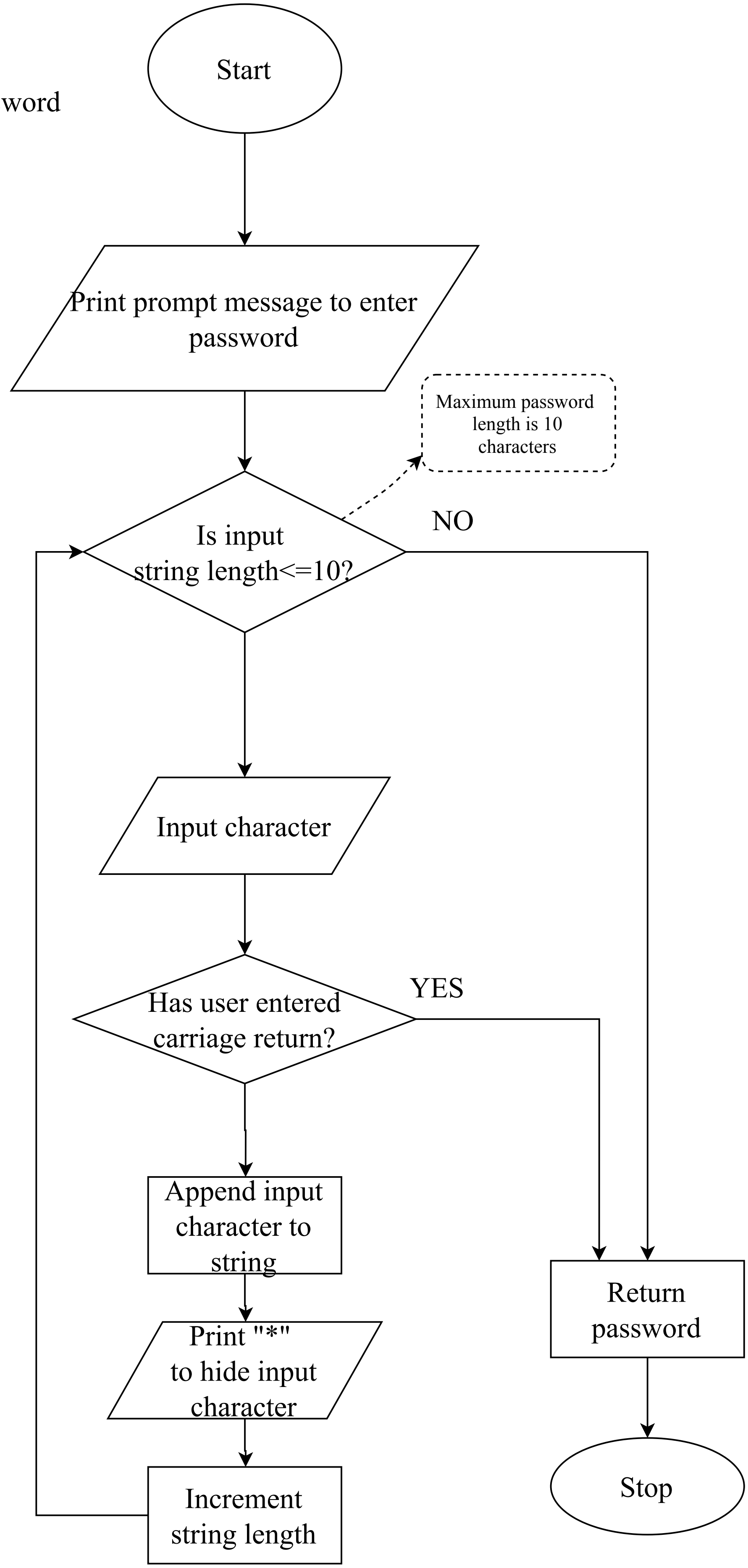

C
A
B
Display all purchase requests (notifications)
Input choice
Does user want to return to main menu?
YES
Display requested notification
Input action to be taken on displayed notification
NO
Is the amount paid?
Delete notification from file

FUNCTION-WISE FLOWCHARTS OF ALL FUNCTIONS USED IN THE PROJECT

Function-1

Prototype:
char* enterpwd (char* str);

This function is for password input, where input is character-wise and password is hidden on the screen

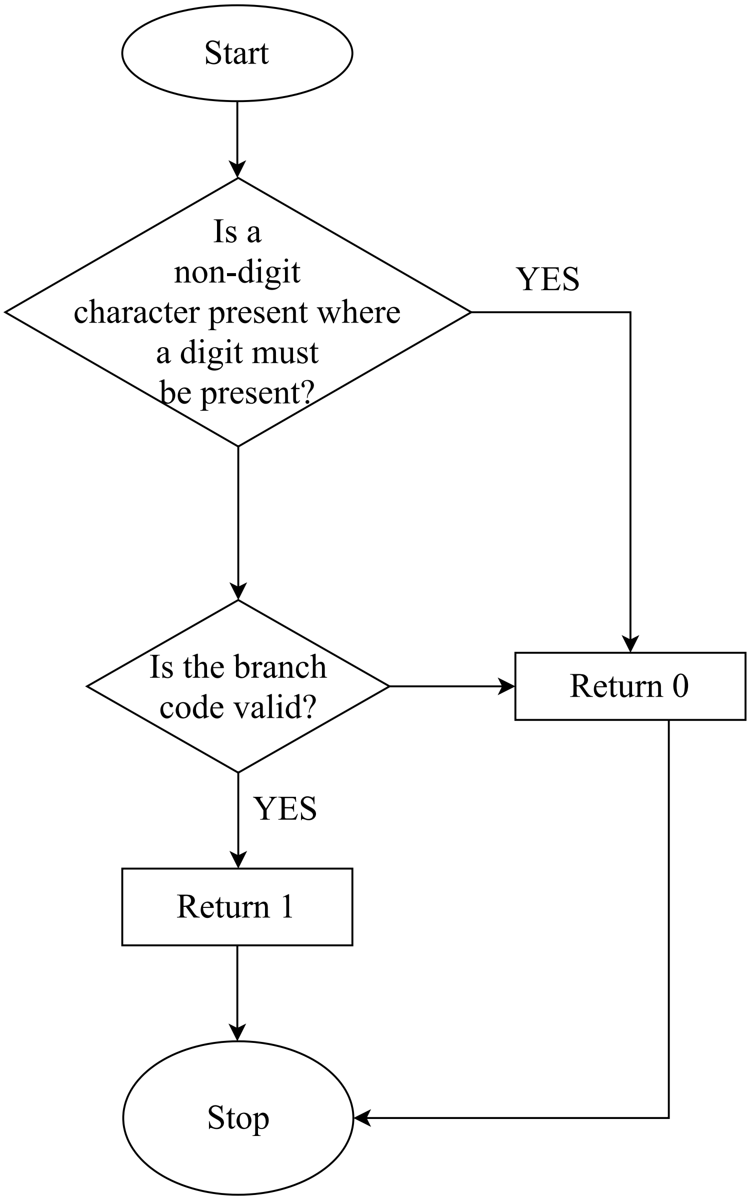

Start
Print prompt message to enter password
Maximum password length is 10 characters
Is input string length<=10?
NO
Input character
Has user entered carriage return?
YES
Append input character to string
Print "*" to hide input character
Increment string length
Return password
Stop

Function-2

Prototype:

int validrno (char *str);

This function validates a student roll number string passed as a parameter.

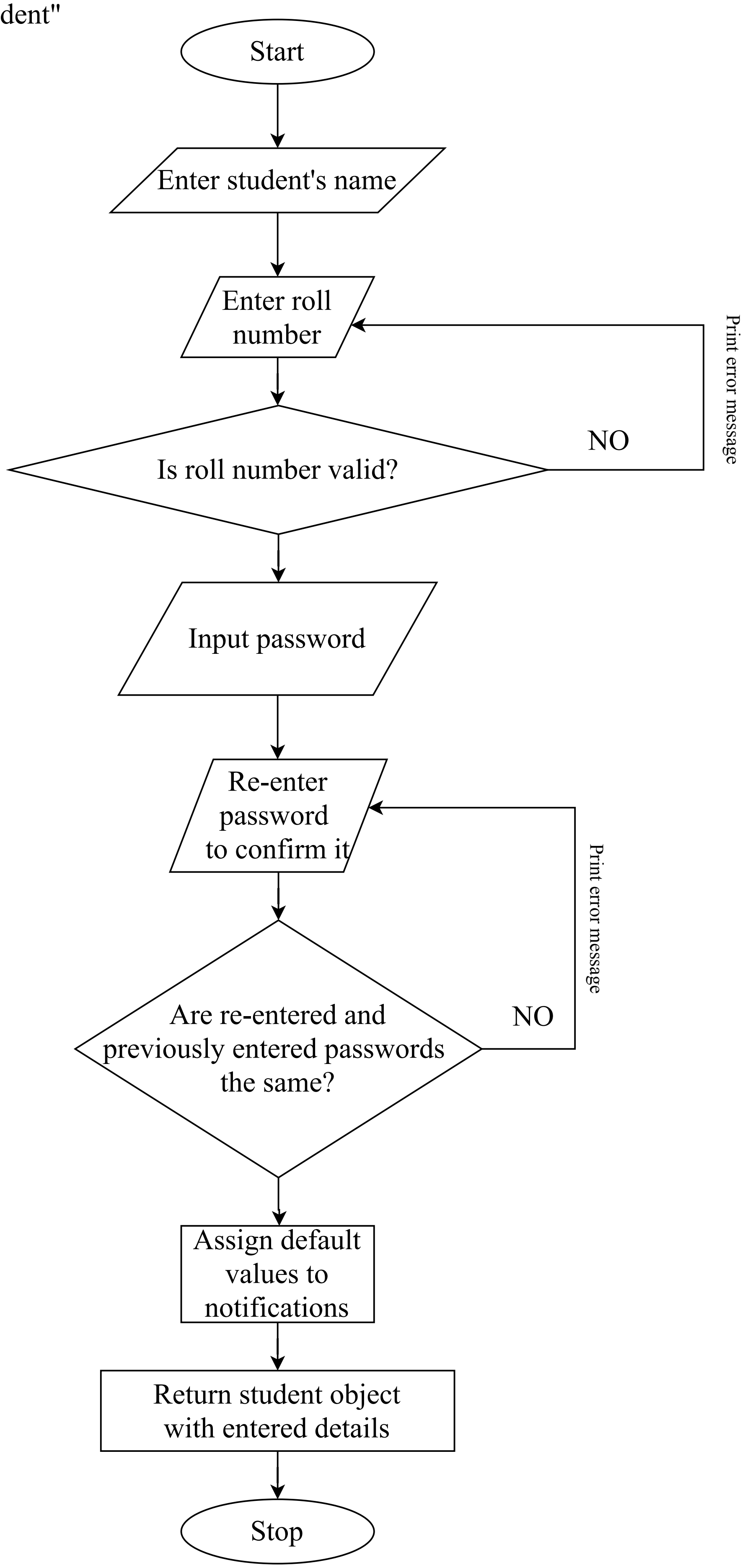

Start
Is a non-digit character present where a digit must be present?
YES
Is the branch code valid?
Return 0
YES
Return 1
Stop

Function-3

Prototype:

```
struct student enter ( struct student s );
```

This function inputs the details of a student for registration, and returns an object of type "student" with the entered details.



Start
Enter student's name
Enter roll number
Is roll number valid?
NO
Print error message
Input password
Re-enter password to confirm it
Are re-entered and previously entered passwords the same?
NO
Print error message
Assign default values to notifications
Return student object with entered details
Stop

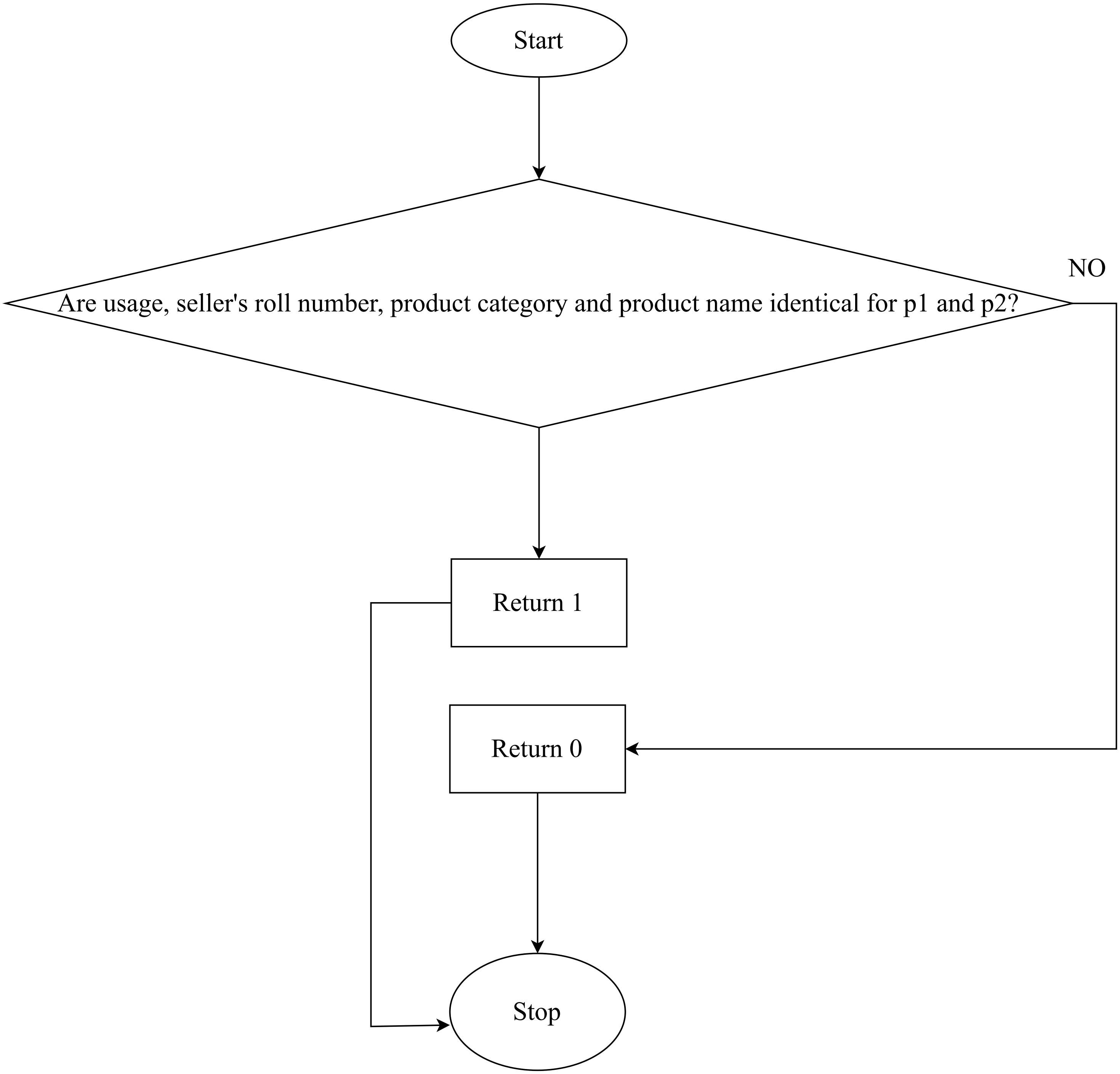
Function - 4

Prototype:

int samepr (struct product p1 , struct product p2);

This function checks if 2 products p1 and p2 are identical.

This will be later used to check if the same student is selling a product twice.

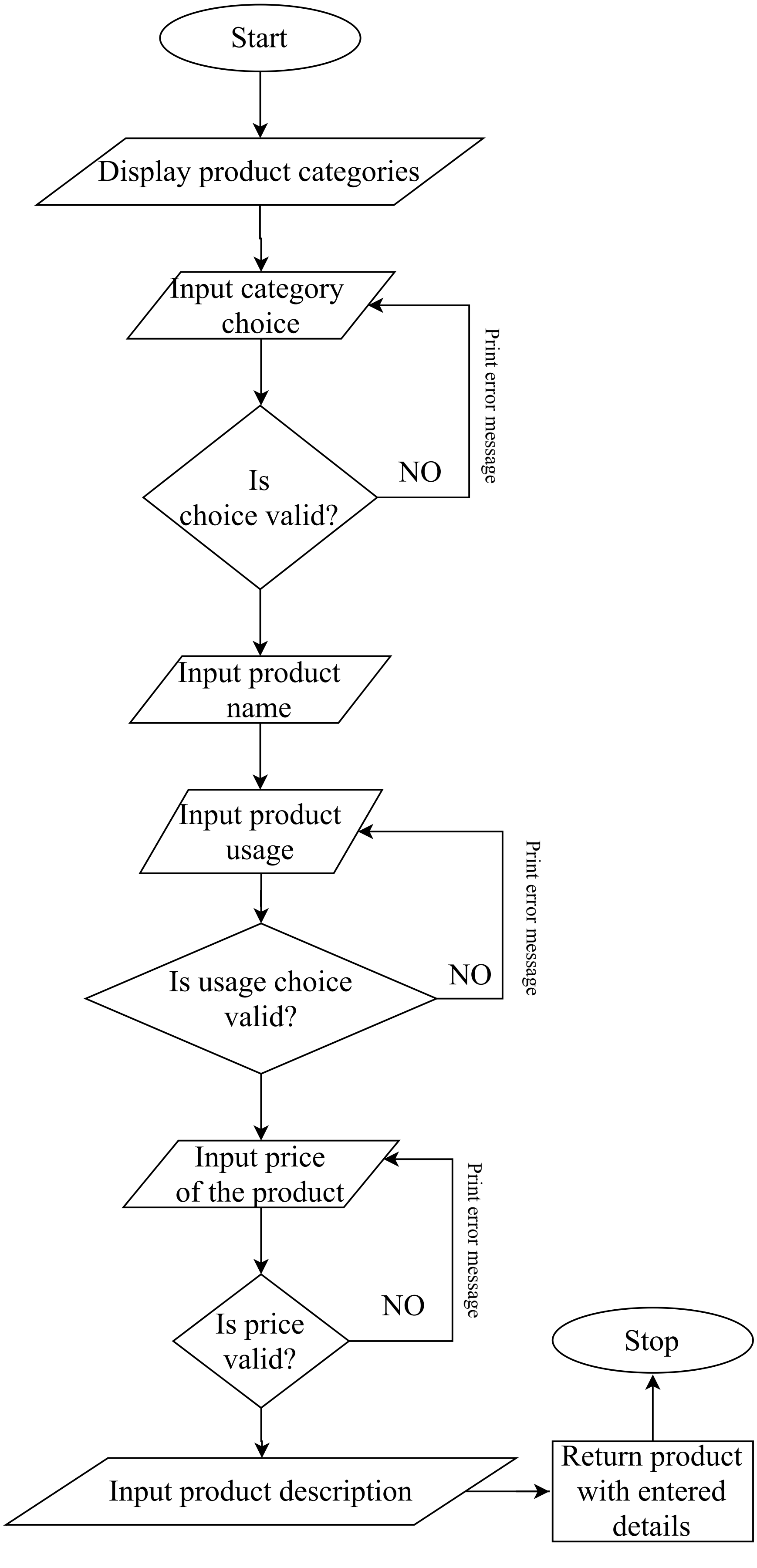

Start
Are usage, seller's roll number, product category and product name identical for p1 and p2?
NO
Return 1
Return 0
Stop

Function - 5

Prototype:

```
struct product enterp ( struct product p );
```

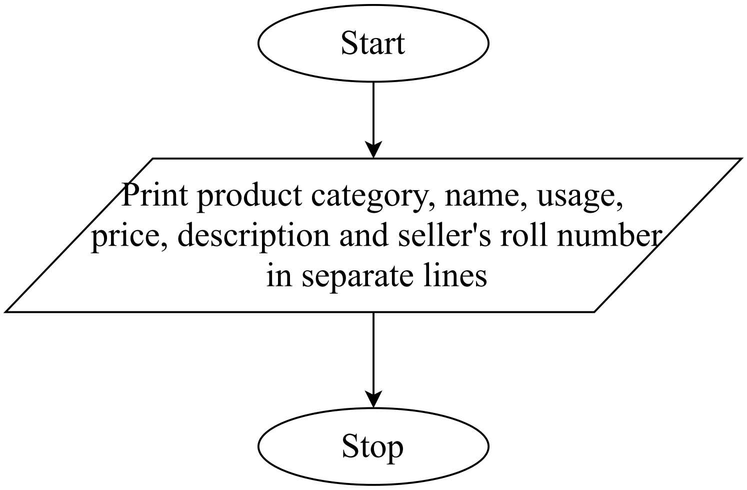
This function enters the details of a product to be put on sale, and returns the data in an object of type "product".



Function - 6

Prototype:
void disppr (struct product p);

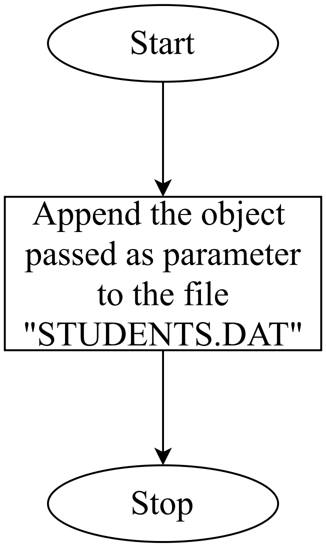
This function displays the details of a product.


Start
Print product category, name, usage, price, description and seller's roll number in separate lines
Stop

Function - 7

Prototype:
void addst (struct student s);

This is used to register a student with the details as a student structure object passed as a parameter.

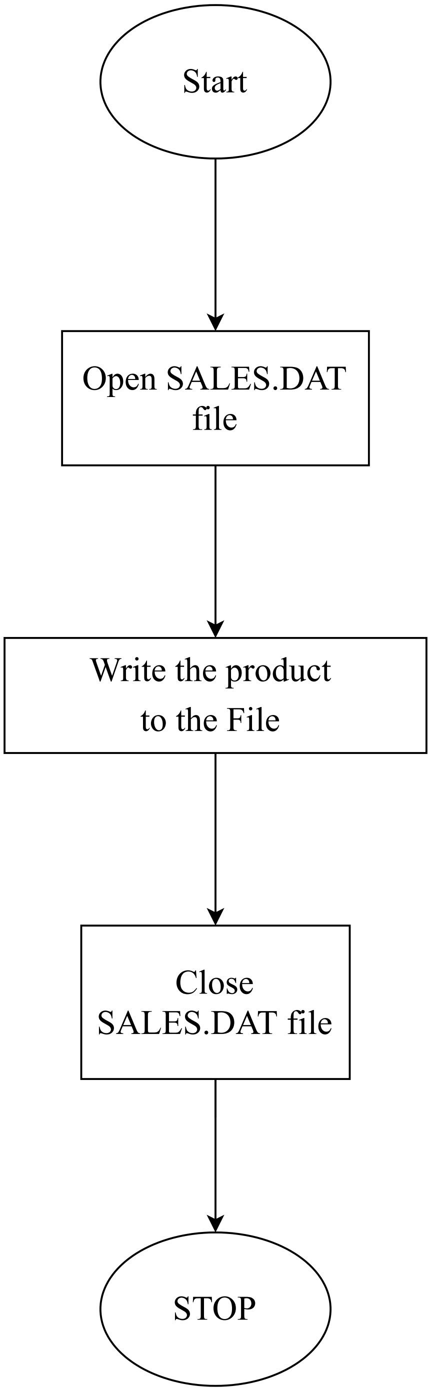

Start
Append the object passed as parameter to the file "STUDENTS.DAT"
Stop

Function - 8

Prototype:

void addpr (struct product p);

Function to add a product to the file "SALES.DAT".

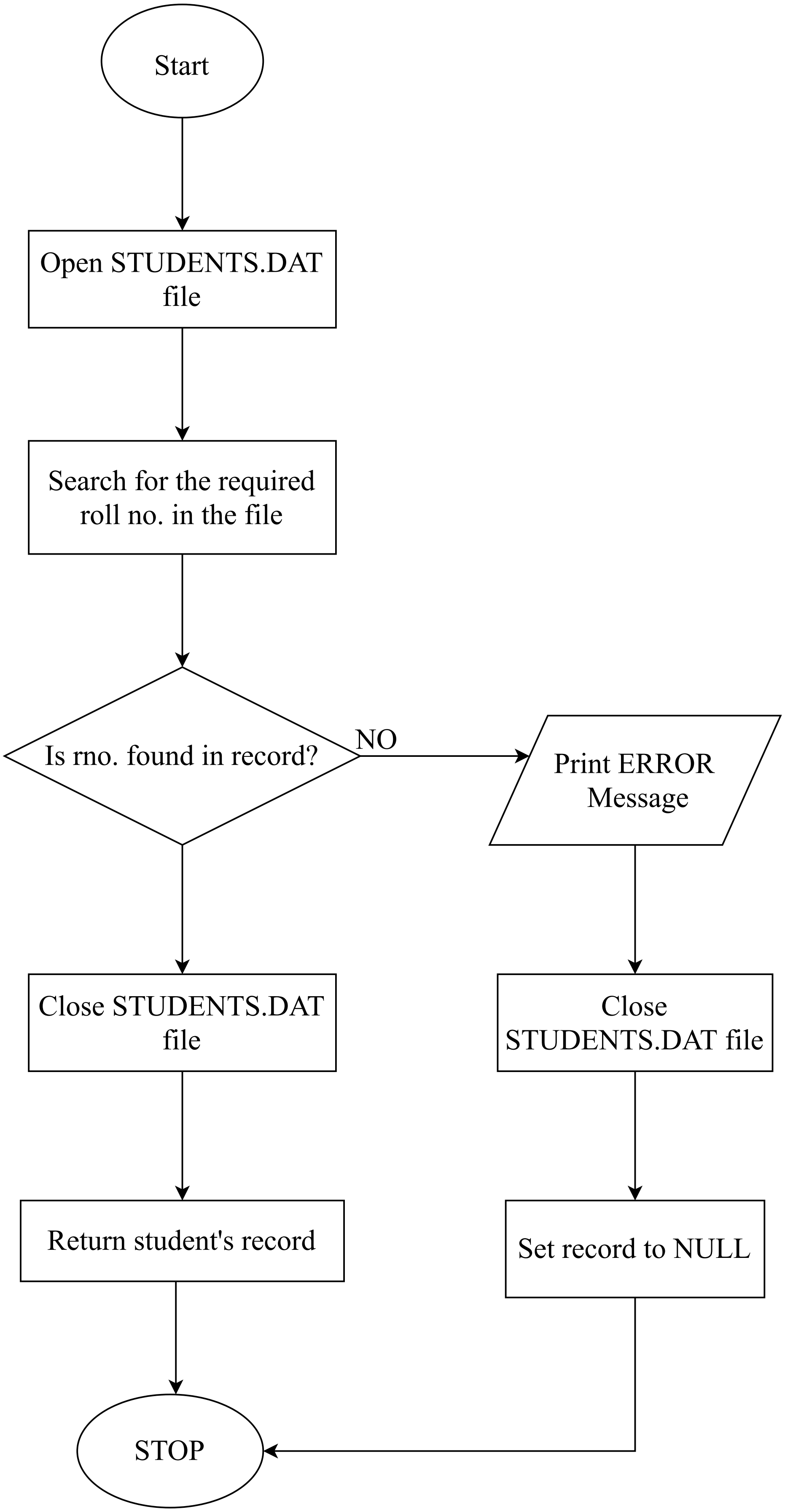


Function - 9

Prototype:

struct student searchrno (char r[10]);

Function to search for a student's record in the file "STUDENTS.DAT", based on roll number.

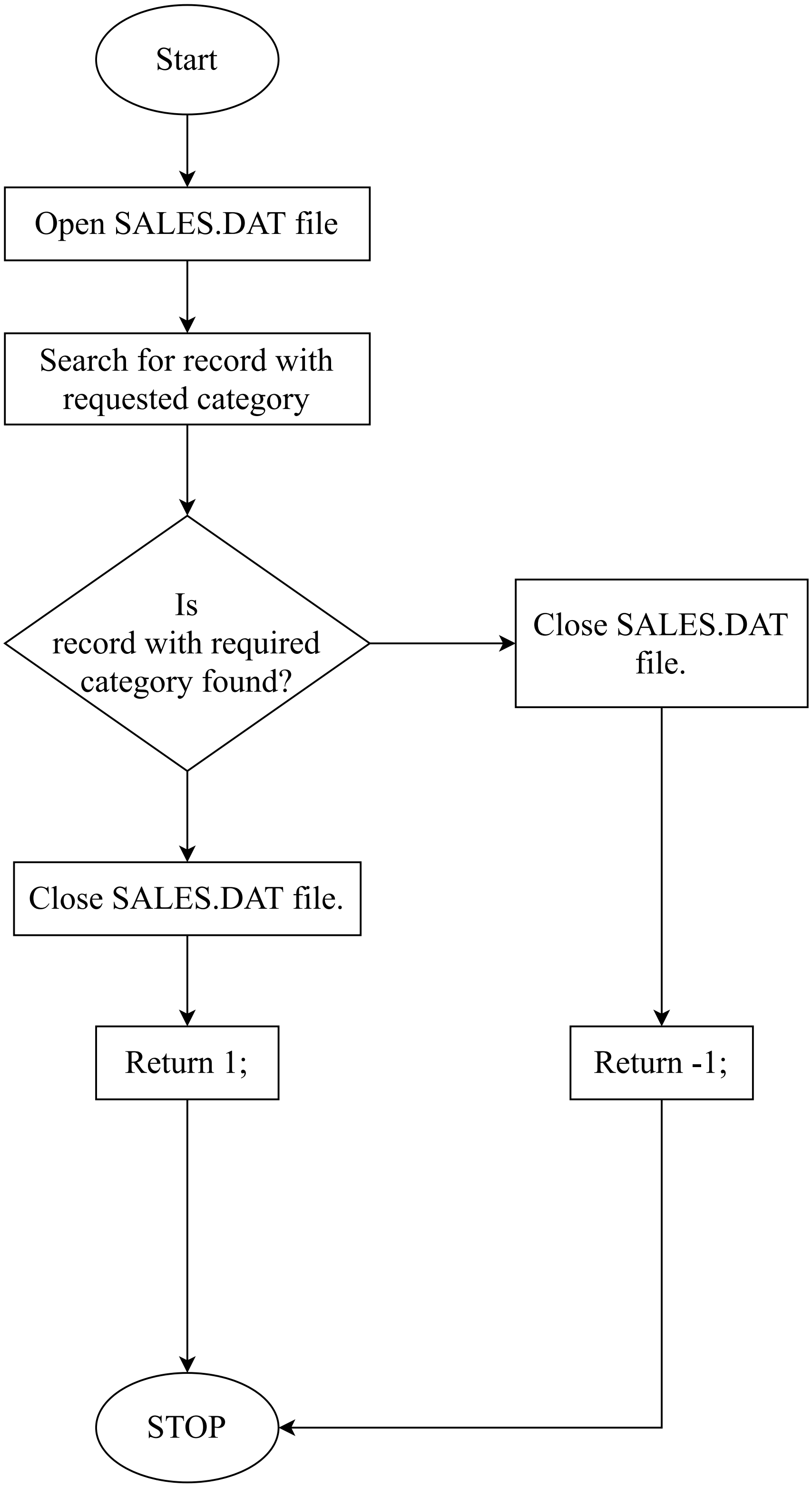


Function - 10

Prototype:

```
int searchcat ( char *c , char *r );
```

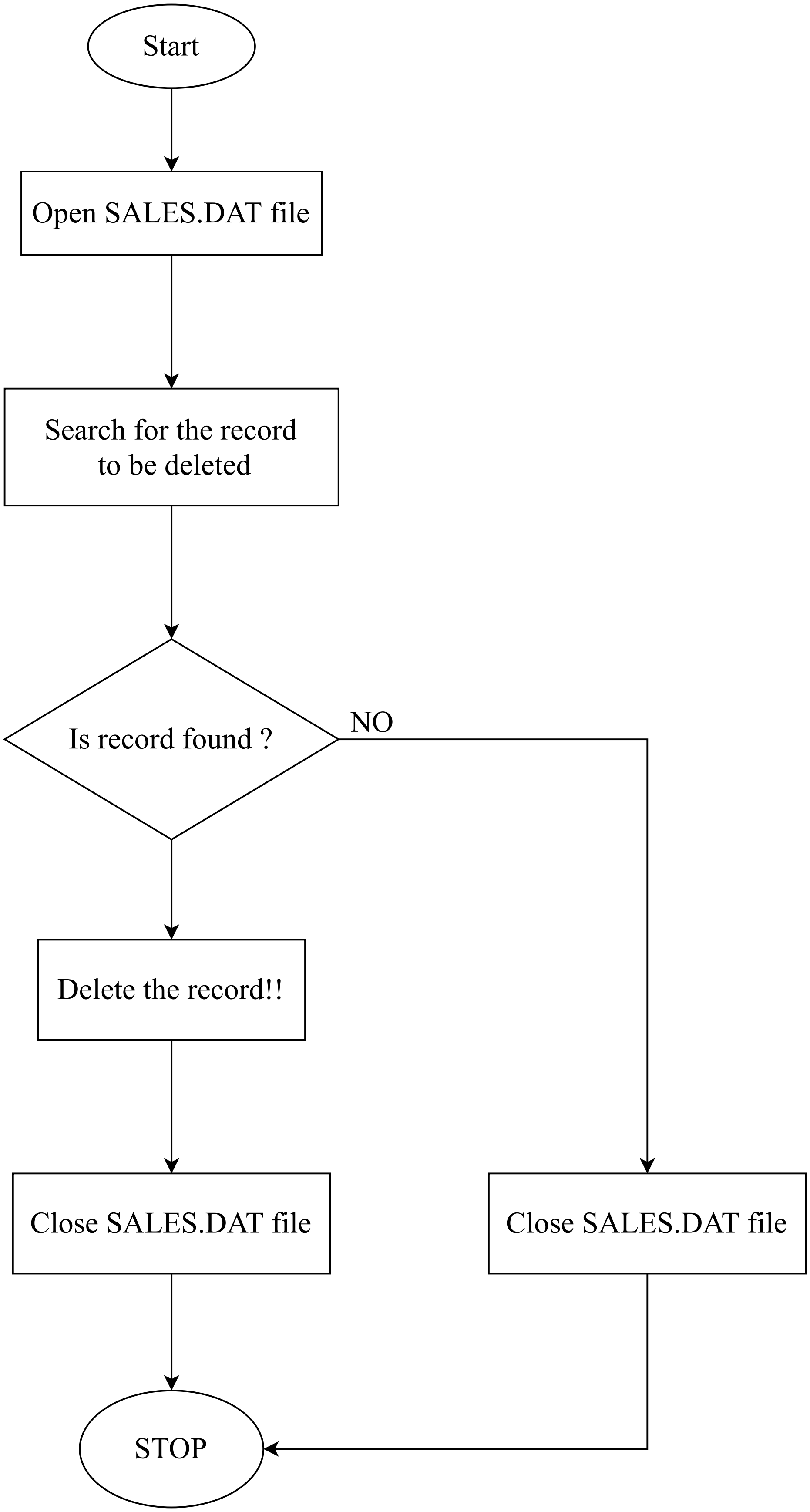
Function for searching a product in the file "SALES.DAT", based on product category.



Function - 11

Prototype:
void delrec (struct product p);

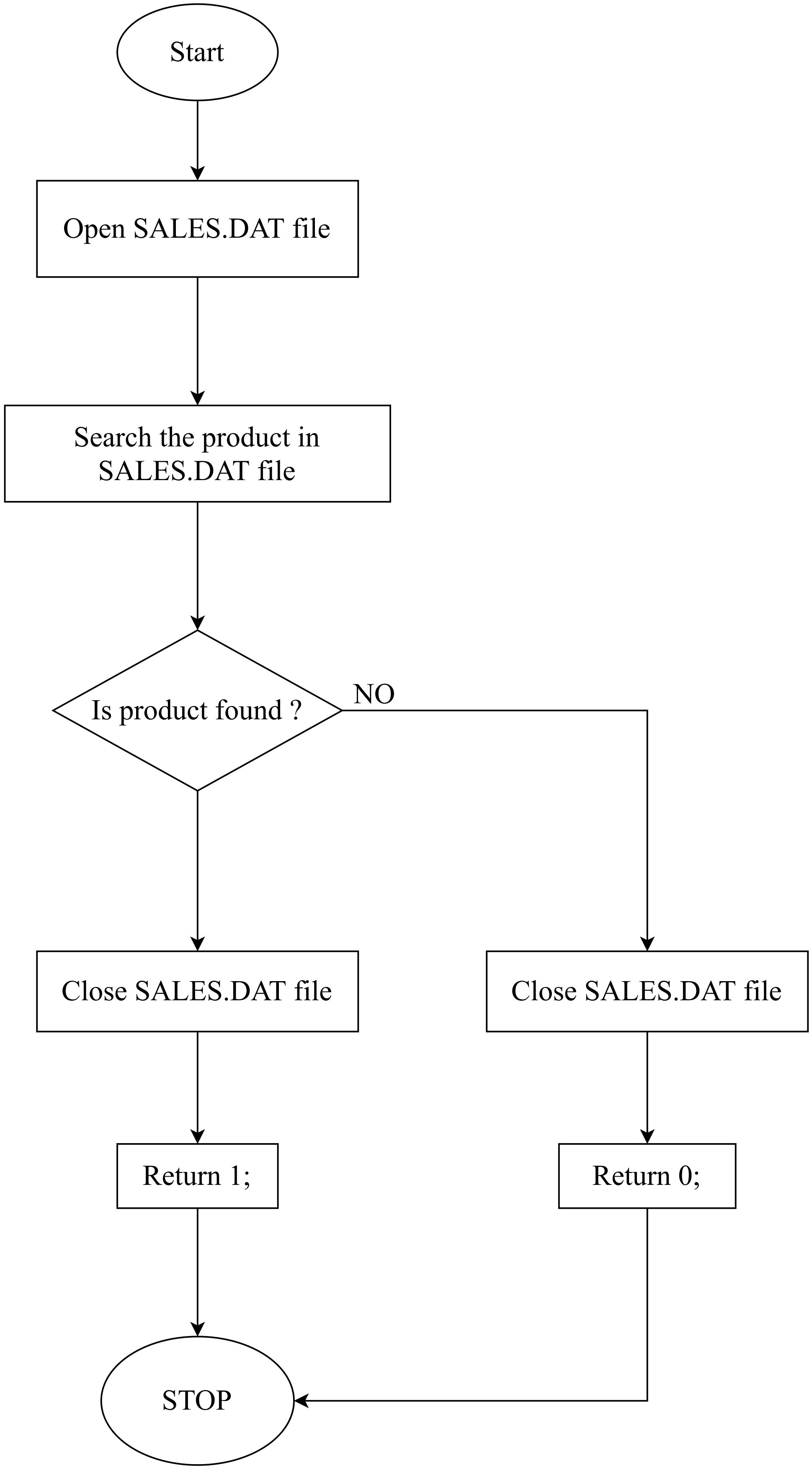
Function to delete the record of a particular product from the file "SALES.DAT"



Function - 12

Prototype:
int searchpr (struct product p);

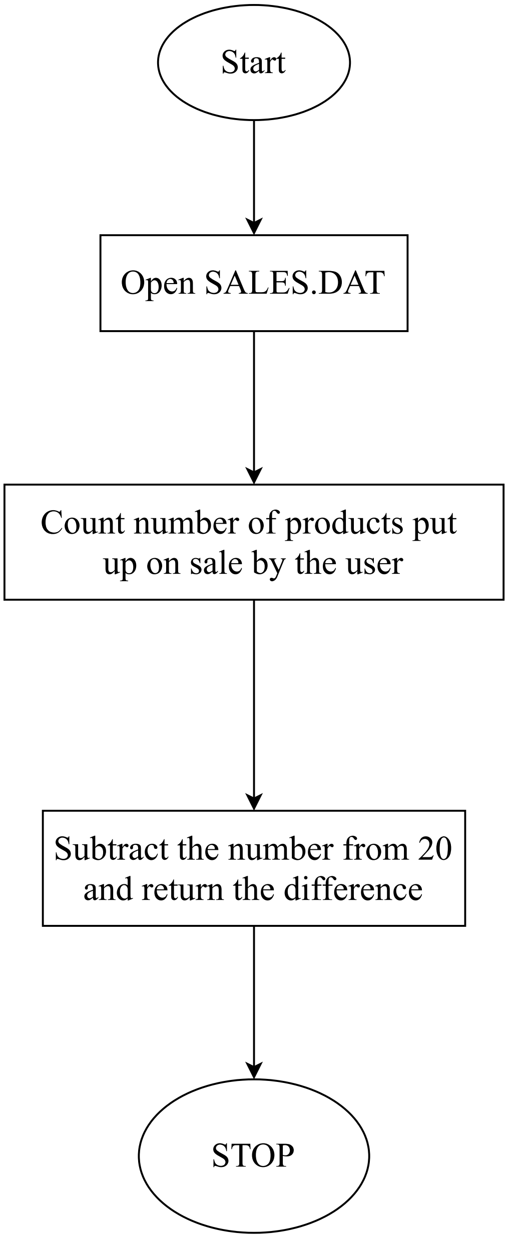
Function to search a product p
in the file "SALES.DAT"



Function - 13

Prototype:
int prodsell(char*);

Function to find the number of products that can still be sold by a particular student.


Start
Open SALES.DAT
Count number of products put up on sale by the user
Subtract the number from 20 and return the difference
STOP