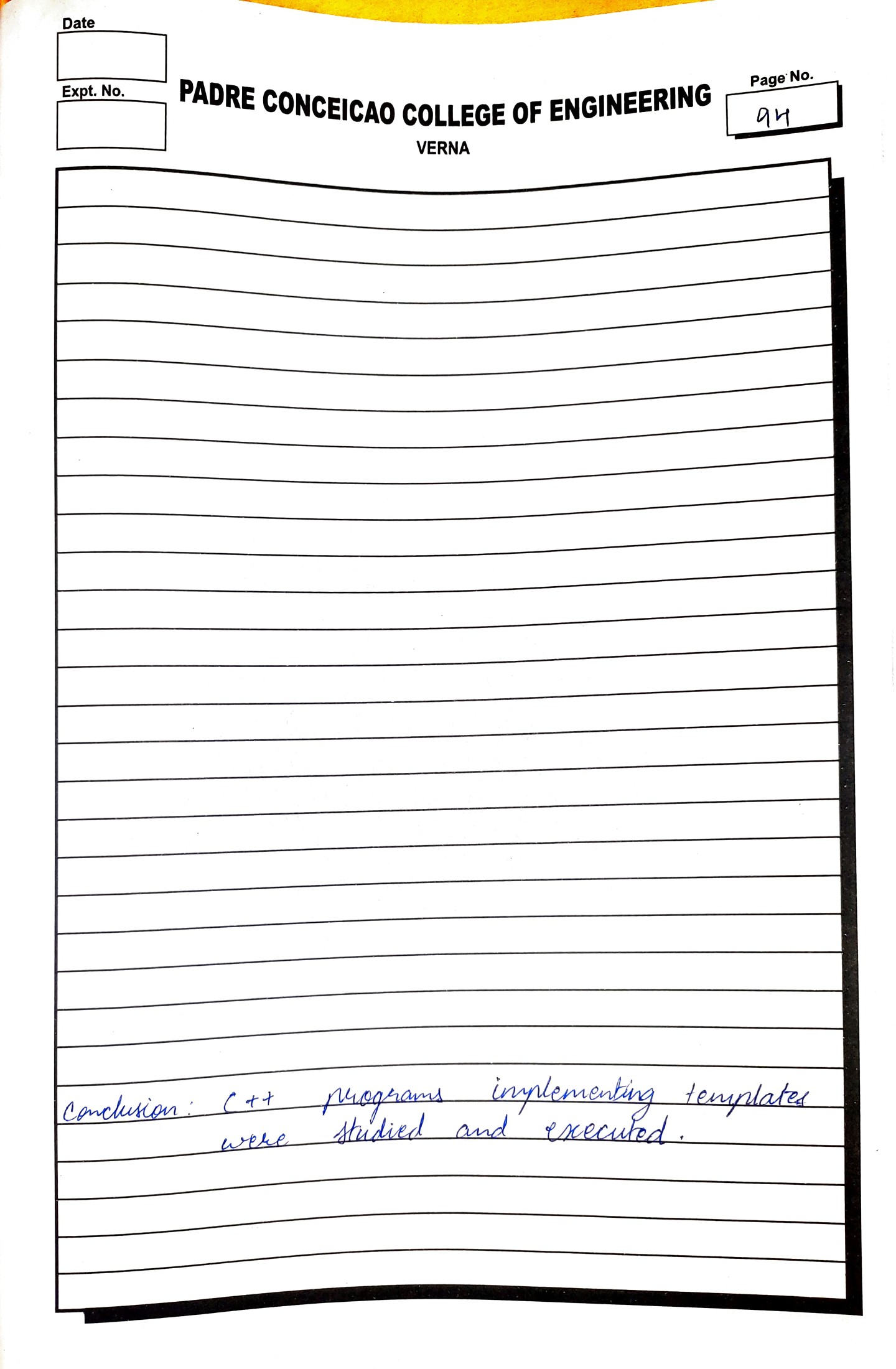
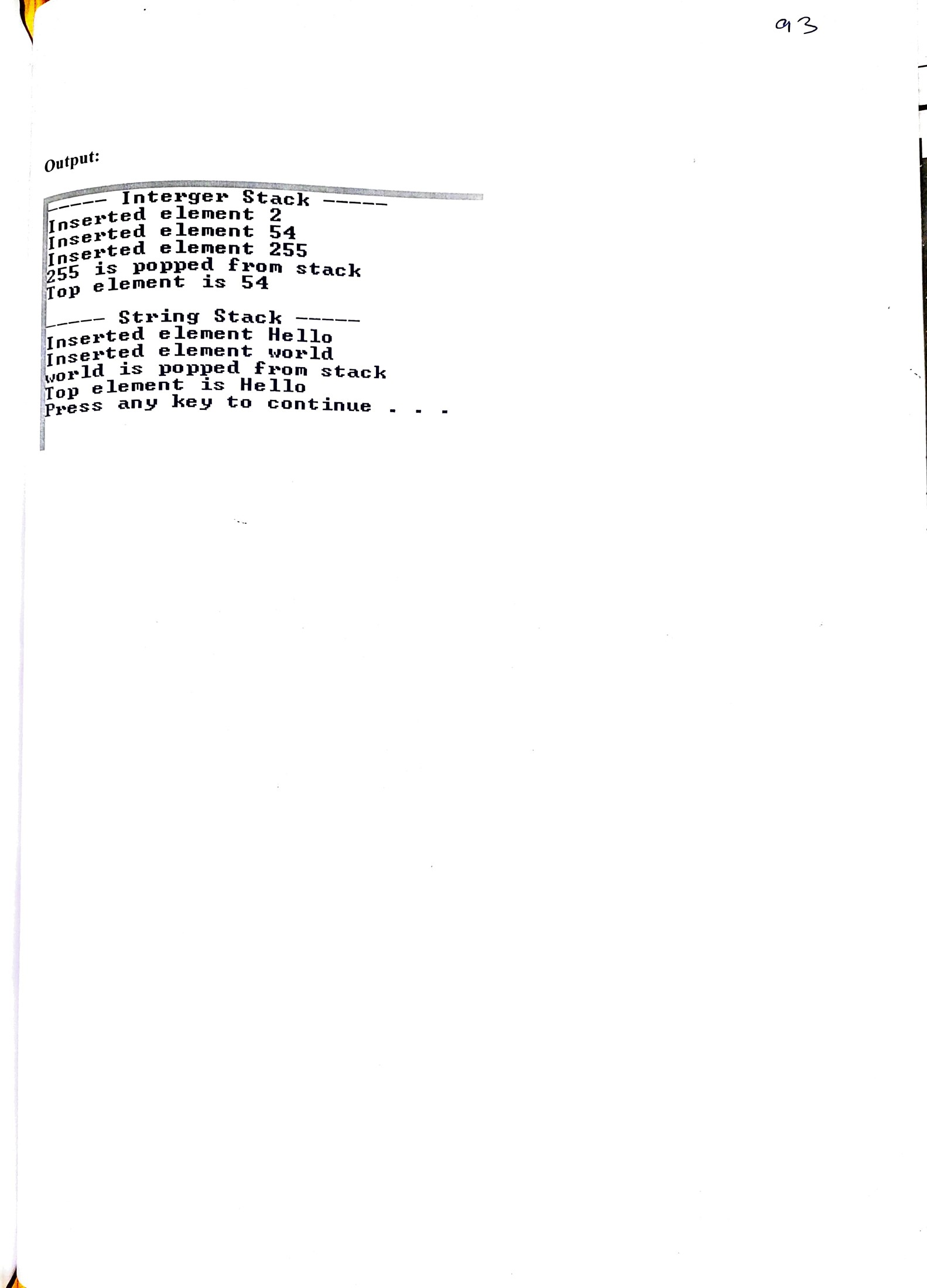
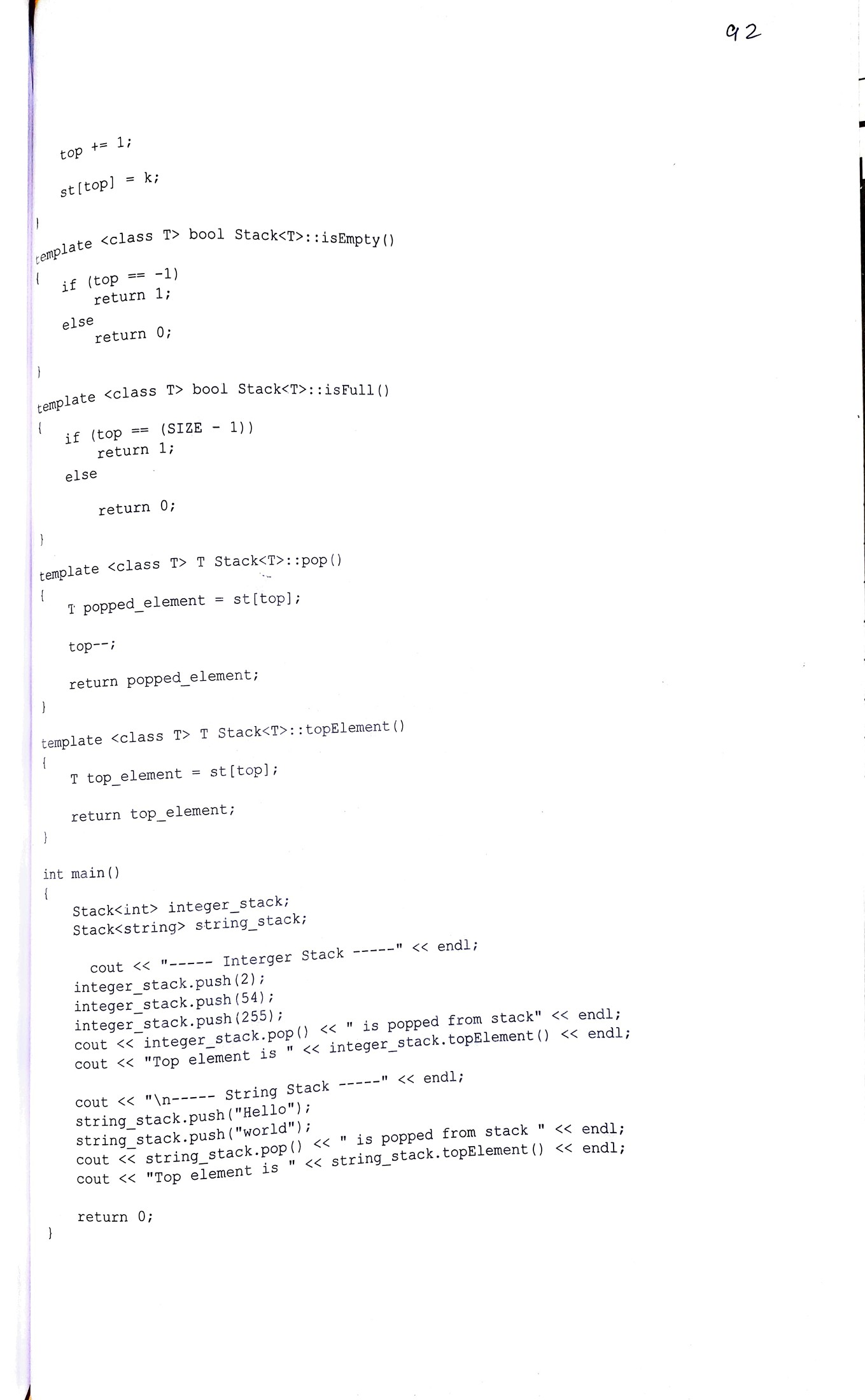
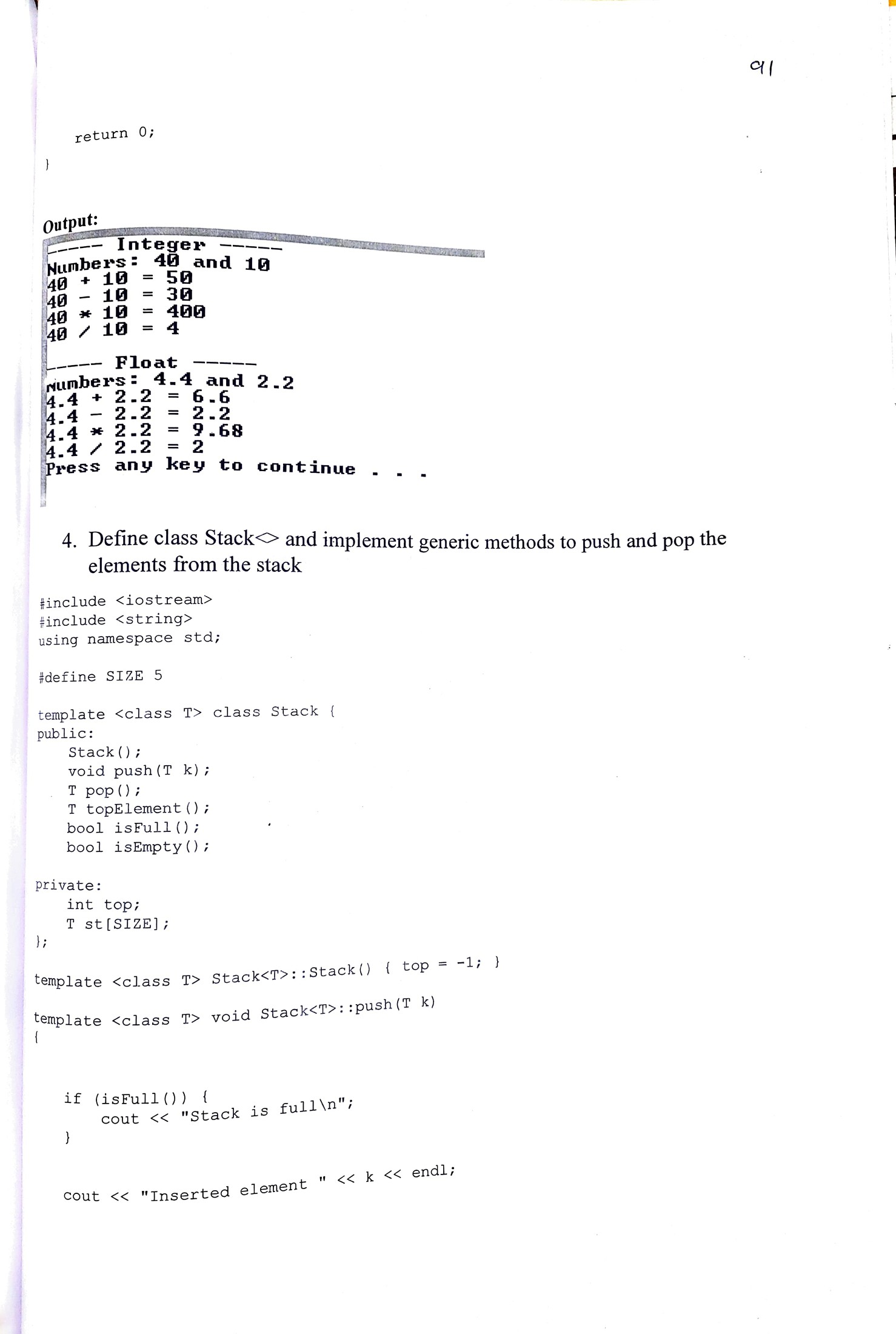
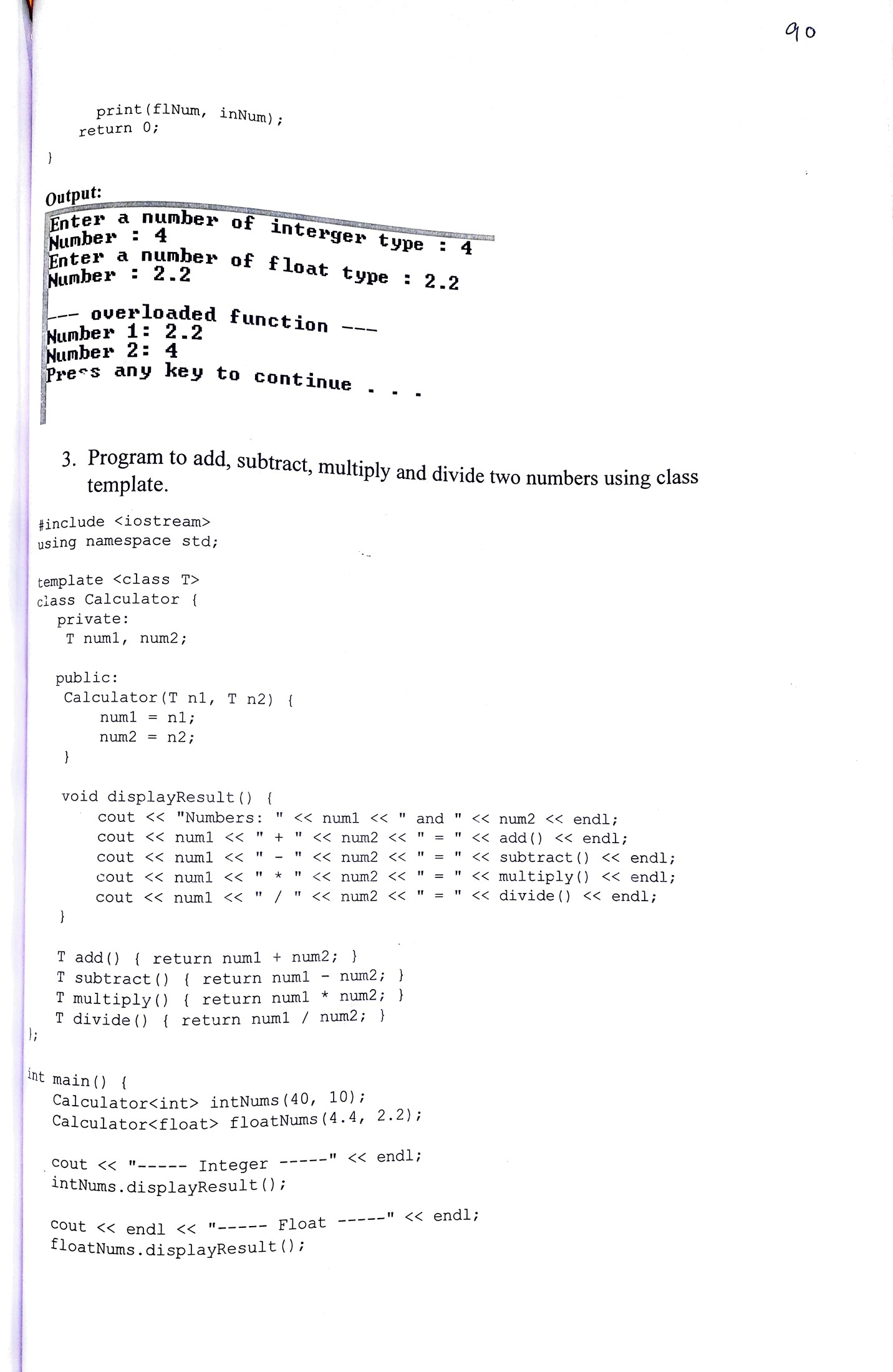
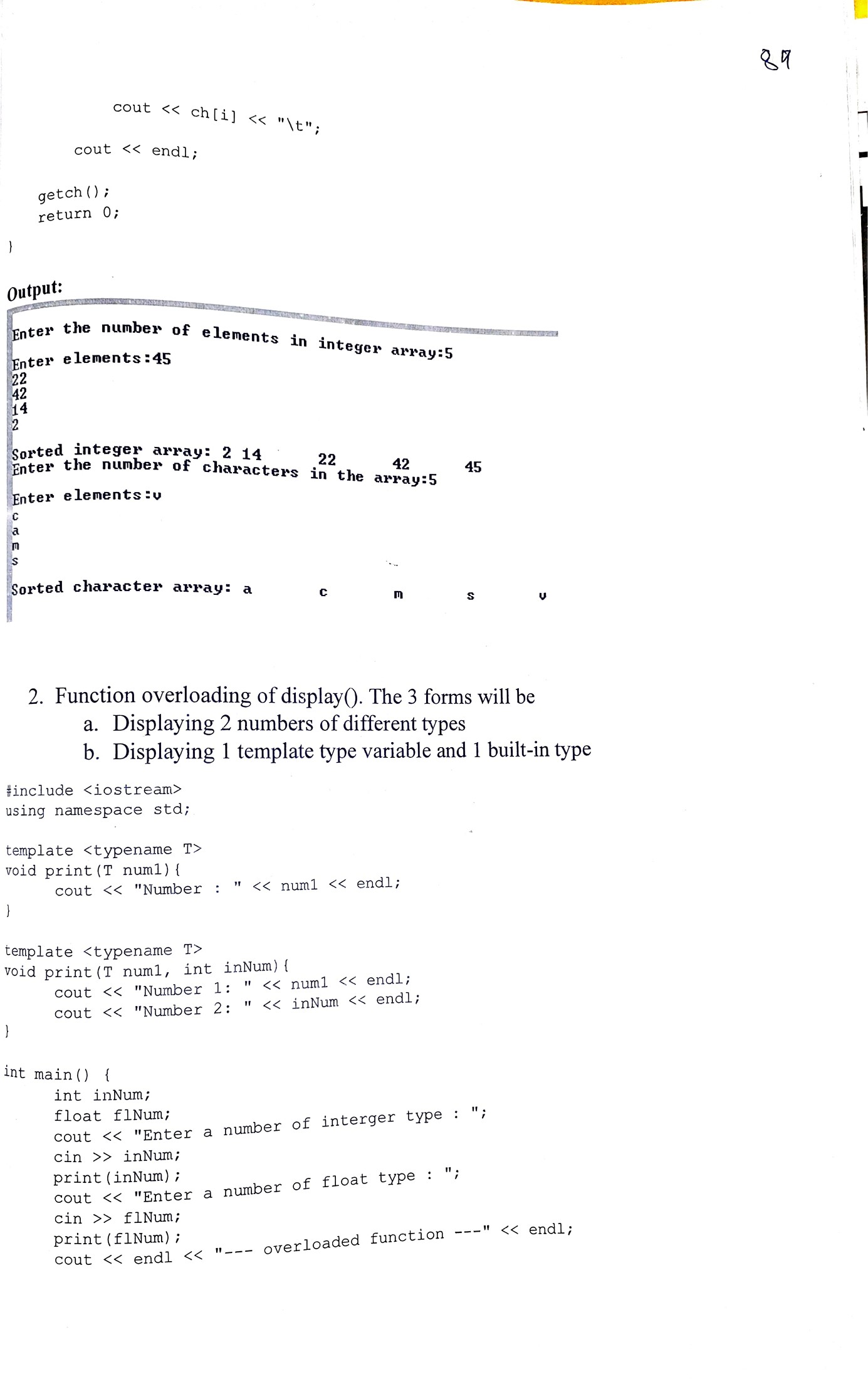
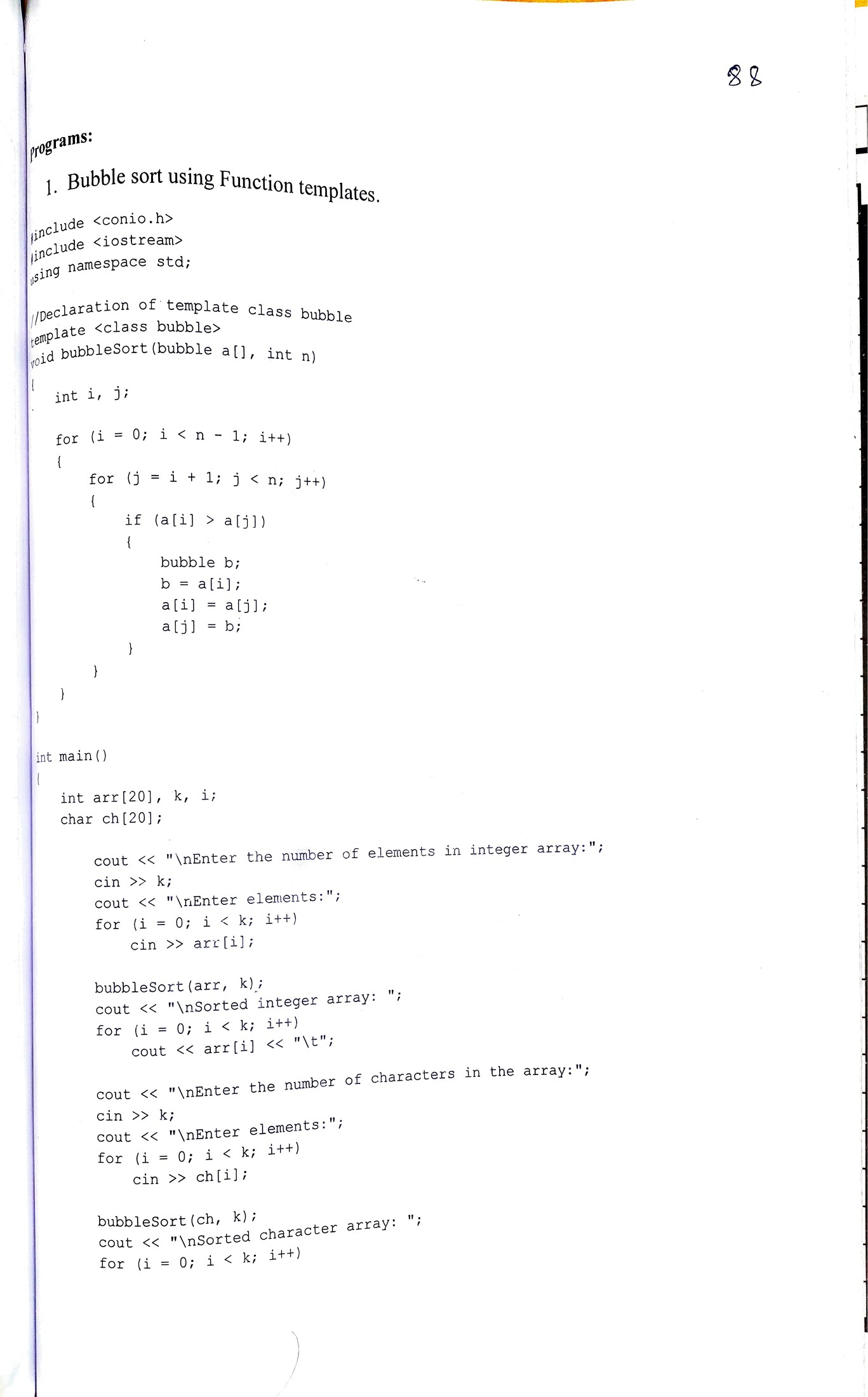
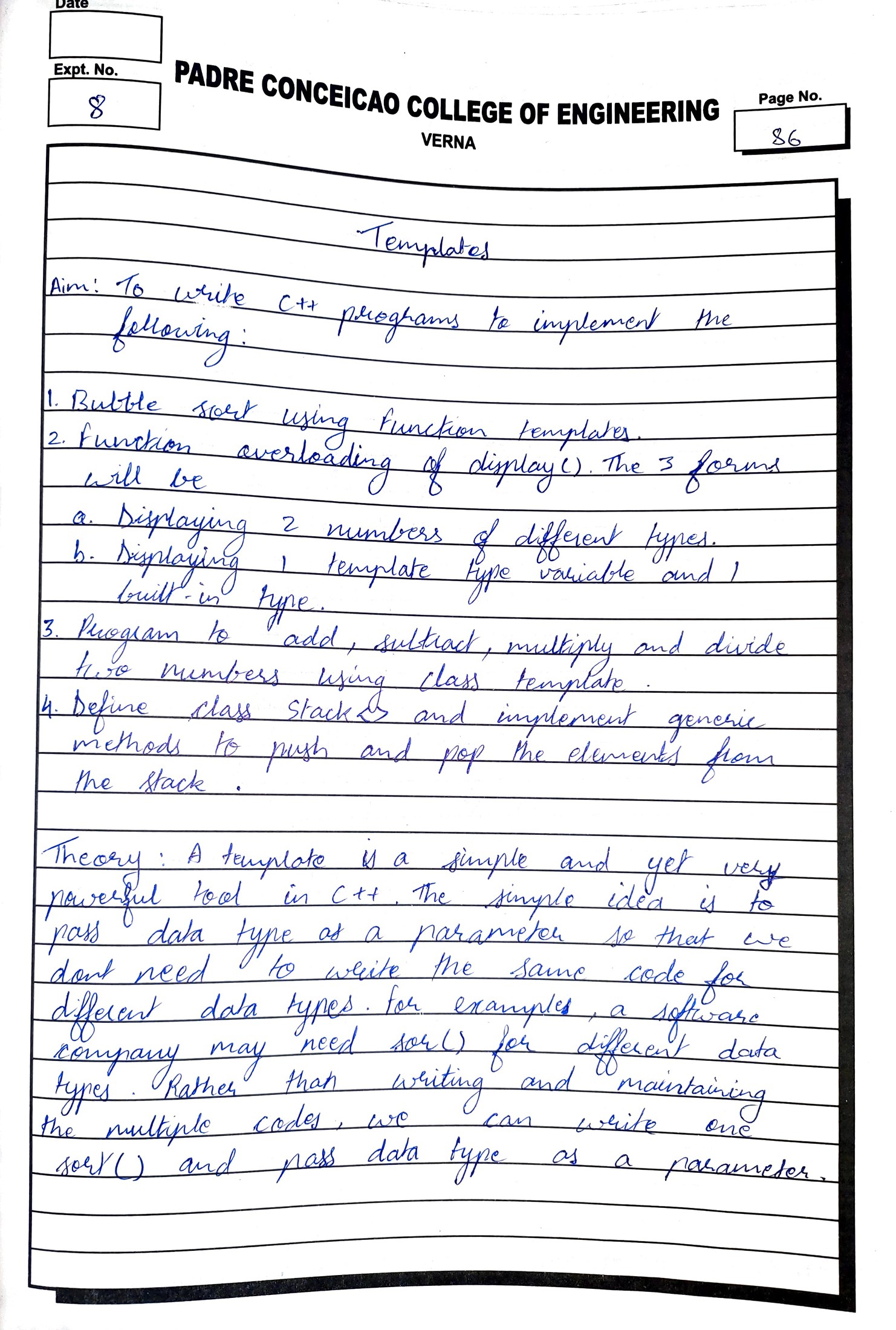
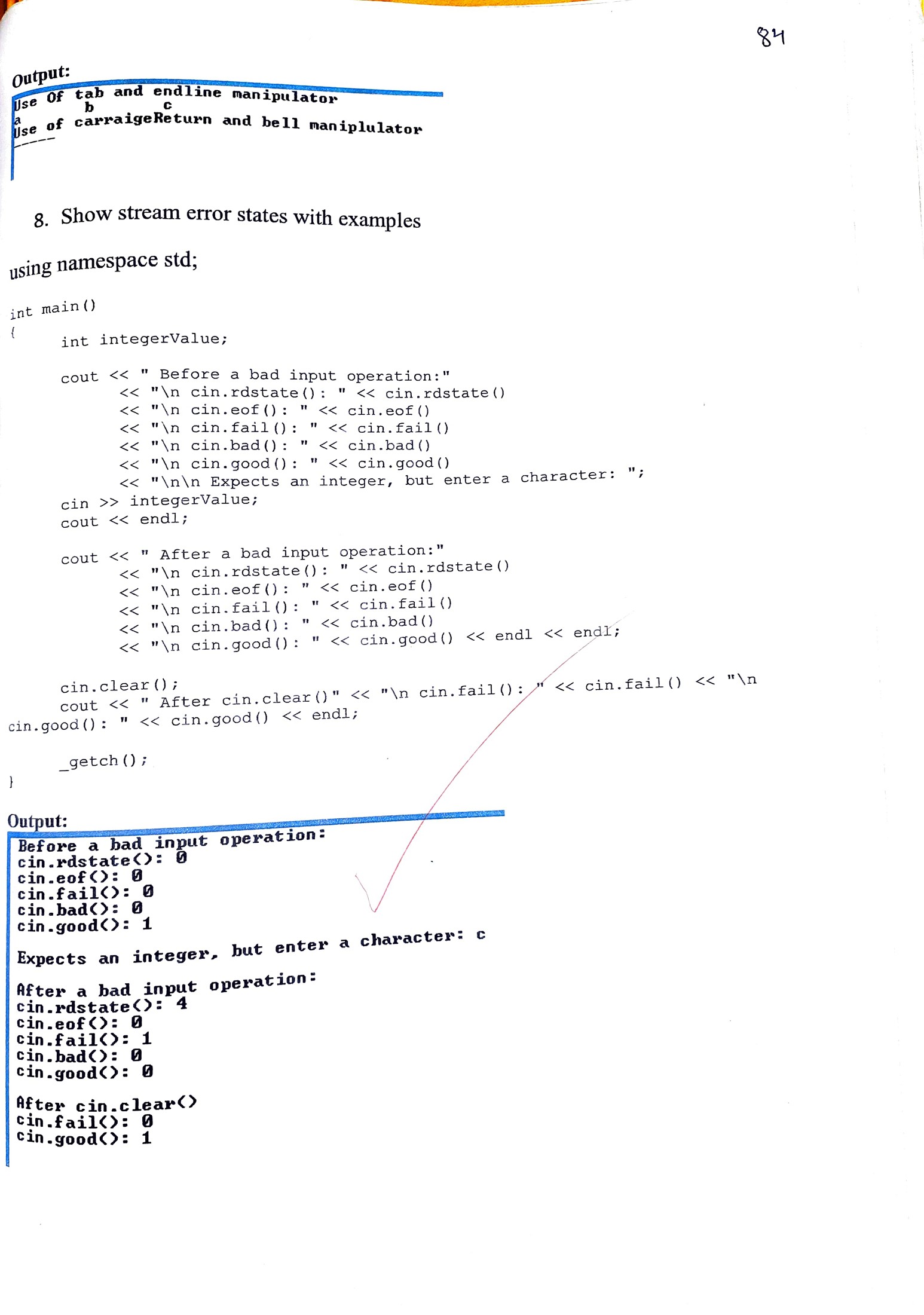
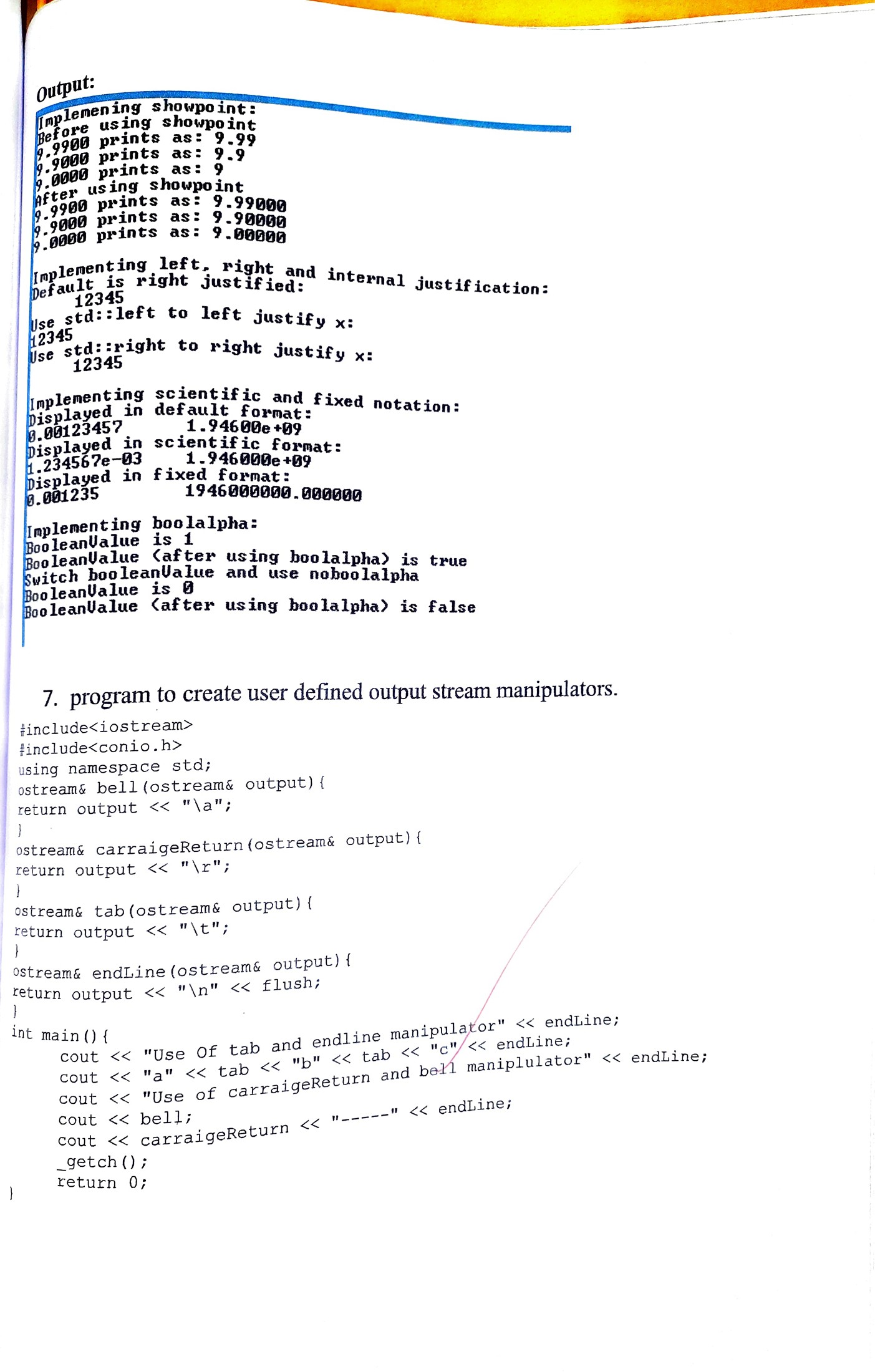
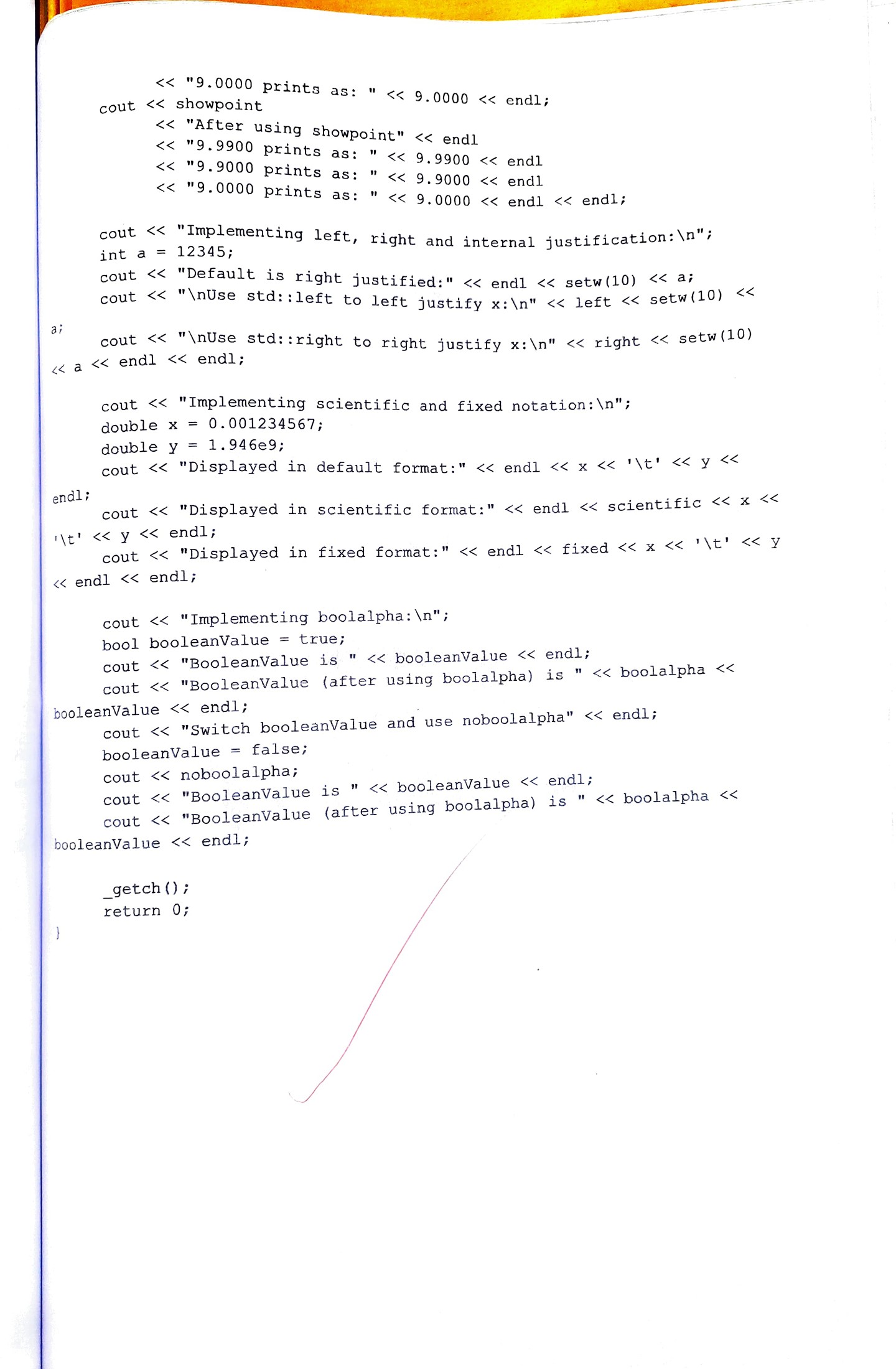
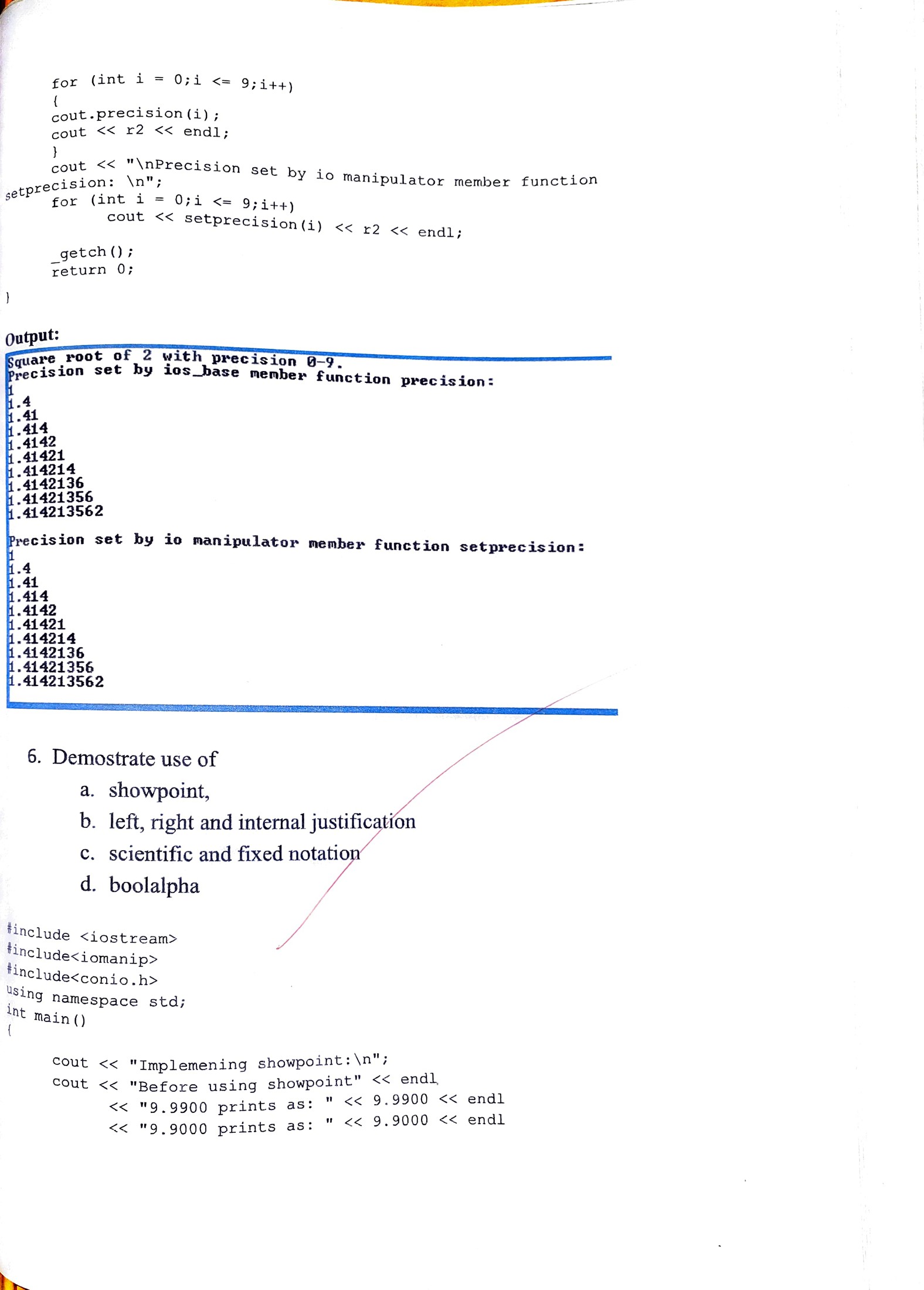
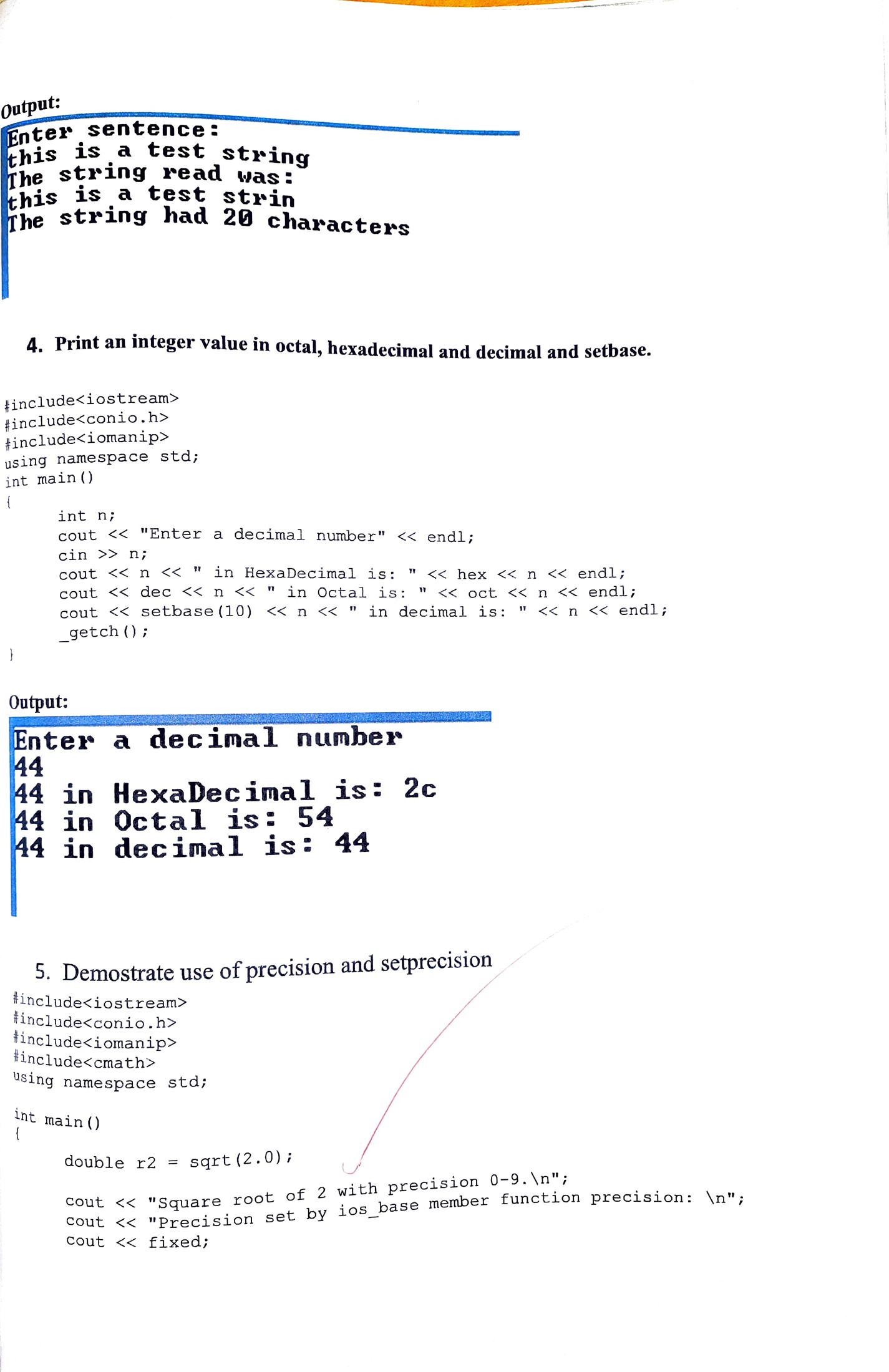
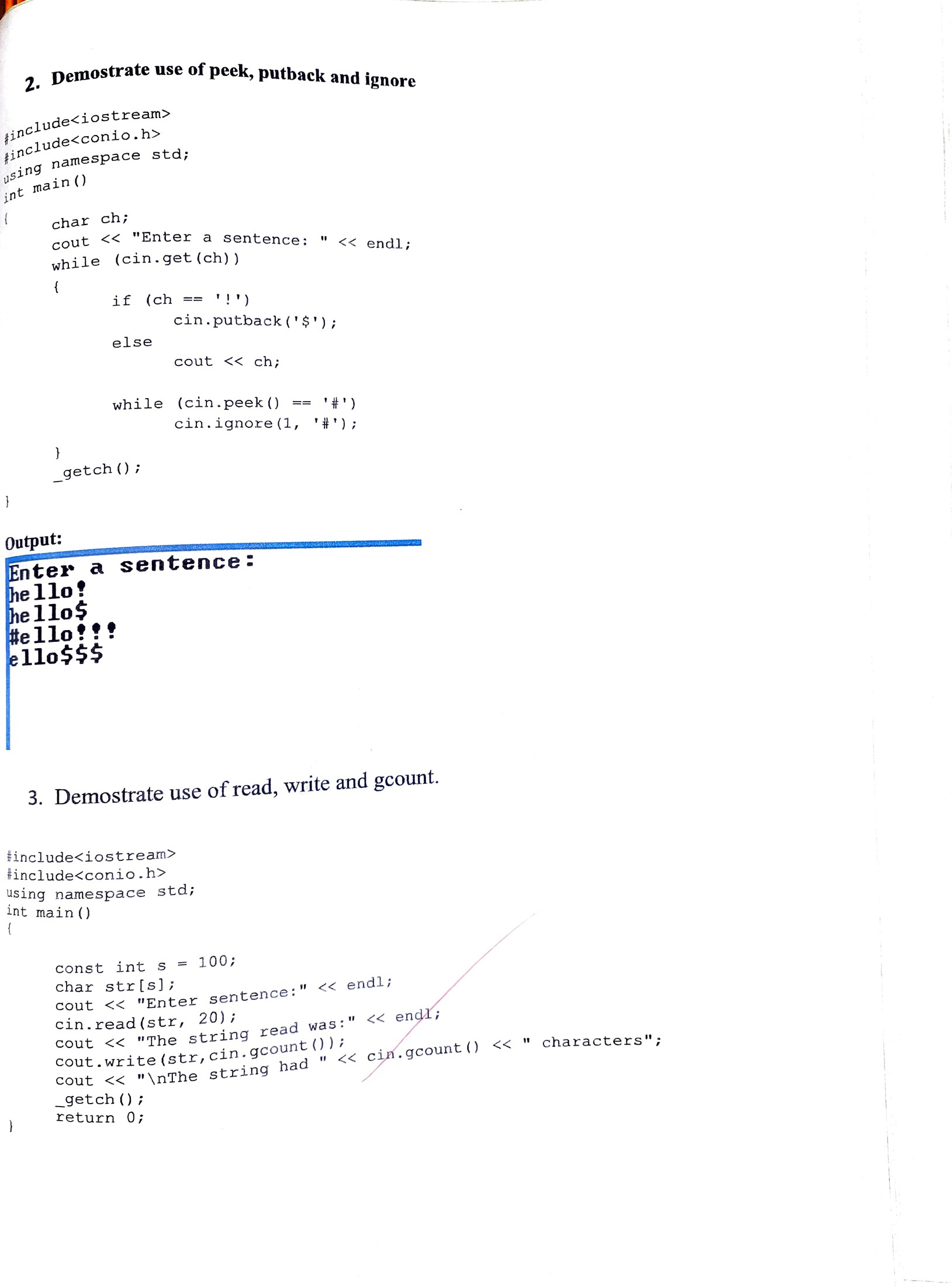
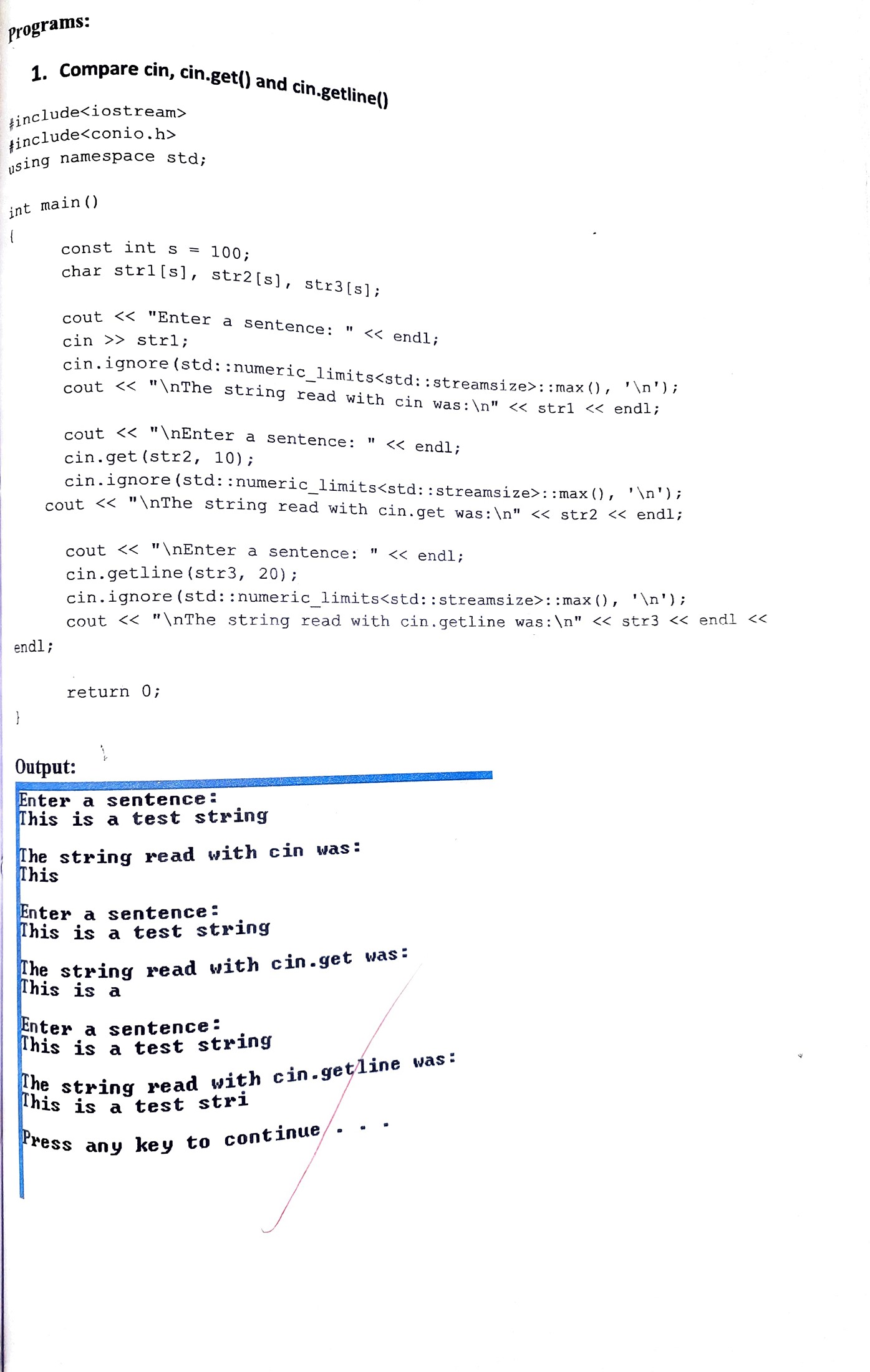
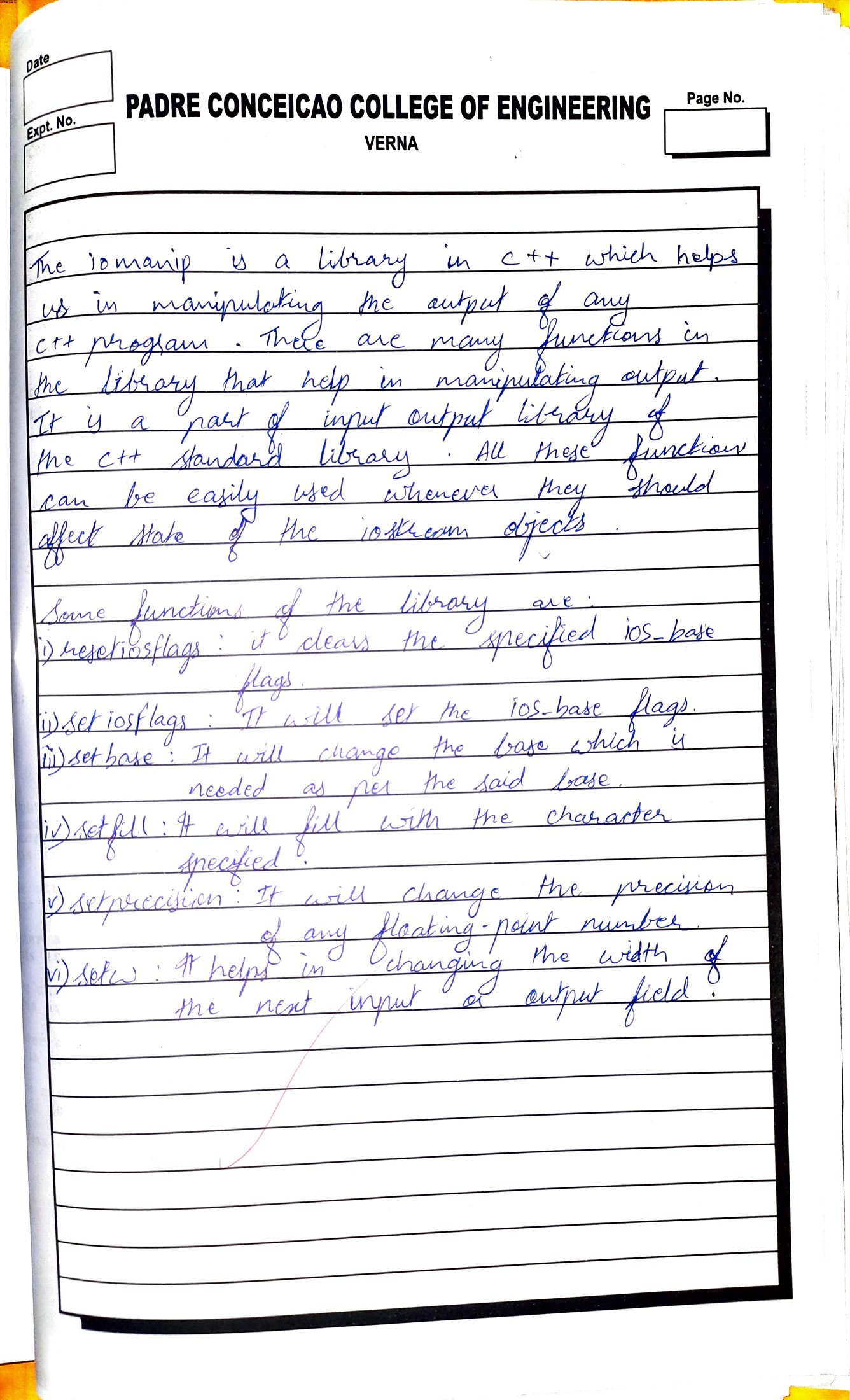
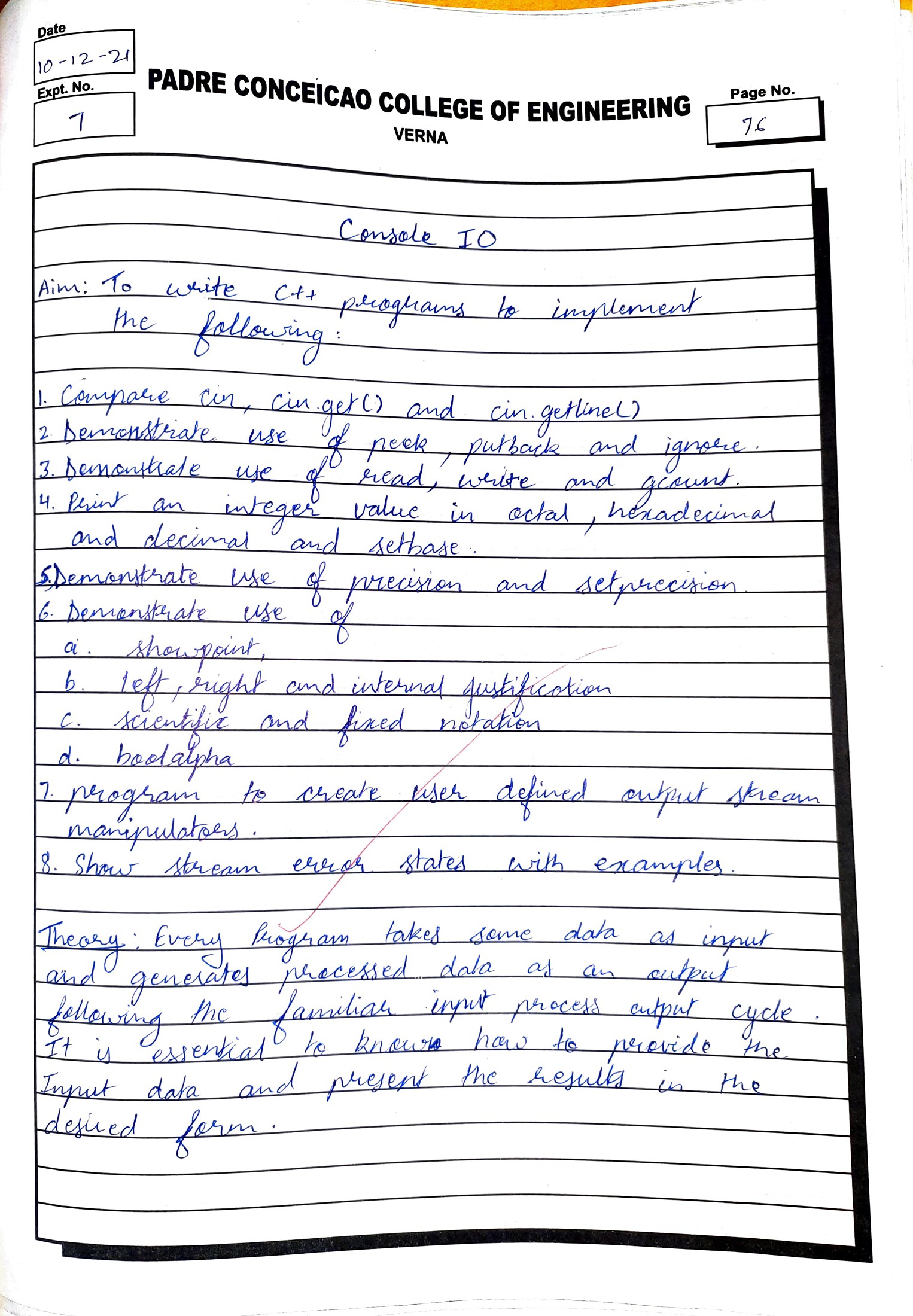
Experiment 7 :



**Mini Project: Travel agency system**



**Program:**

#include<iostream>

#include<conio.h>

#include<fstream>

#include<sstream>

#include<string>

#include<cstdlib>

#include<windows.h>

#include<iomanip>

#include<time.h>

using namespace std;

const int LINELENGTH = 80;

enum Position{ LEFT, CENTRE, RIGHT };

void clearScreen()

{

system("cls");

cout << "Click 'ctrl + c' to exit the program." << endl;

}

void wait(int time = 800)

{

Sleep(time);

}

int generateRandomId(){

wait();

srand(time(0));

return rand();

}

void print( Position pos, string s, int linelength )

{

int spaces = 0;

switch( pos )

{

case CENTRE: spaces = ( linelength - s.size() ) / 2; break;

case RIGHT : spaces = linelength - s.size() ; break;

}

if ( spaces > 0 ) cout << string( spaces, ' ' );

cout << s << '\n';

}

void print(int LINELENGTH){

string border(LINELENGTH, '=');

cout << border << endl;

print( CENTRE, "Welcome To Travel Agency System", LINELENGTH );

cout << endl << border << endl;

}

void printMainMenu(int LINELENGTH){

print(CENTRE, "Travel Agency Menu", LINELENGTH);

cout << "\n 1. Plan new trip." << endl;

cout << " 2. Cancel Trip." << endl;

cout << " 3. Exit." << endl;

cout << "\n Choose One: ";

}

class Transport{

protected:

int transport\_id;

string transport\_type;

int transport\_cost;

public:

Transport(){

transport\_id = 0;

transport\_type = "";

transport\_cost = 0;

}

void getTransportDetails(int tripOption){

// assigning id

transport\_id = generateRandomId();

//input transport type

cout << "\n Enter(exact word) transport type from the following " << endl;

string border(LINELENGTH,'\*');

cout << endl << border;

cout << "|" << left << setw(20) << " " << " | " << left << setw(55) << "Transport type" << right << "|";

cout << border;

cout << "|" << left << setw(20) << " 1" << " | " << left << setw(55) << "Bus" << right << "|";

cout << "|" << left << setw(20) << " 2" << " | " << left << setw(55) << "Flight" << right << "|";

cout << "|" << left << setw(20) << " 3" << " | " << left << setw(55) << "Train" << right << "|";

cout << "|" << left << setw(20) << " 4" << " | " << left << setw(55) << "Cab" << right << "|";

cout << border << endl;

cout << ": ";

getline(cin, transport\_type);

// assigning the transport cost

if(transport\_type == "Bus" || transport\_type == "bus" || transport\_type == "BUS"){

transport\_cost = 1200 \* tripOption;

}else if(transport\_type == "Flight" || transport\_type == "flight" || transport\_type == "BUS"){

transport\_cost = 4500 \* tripOption;

}else if(transport\_type == "Train" || transport\_type == "train" || transport\_type == "TRAIN"){

transport\_cost = 900 \* tripOption;

}else if(transport\_type == "Cab" || transport\_type == "cab" || transport\_type == "CAB"){

transport\_cost = 4000 \* tripOption;

}else {

transport\_cost = 3000;

}

}

void bookTransport(int trip\_id){

ofstream fout;

fout.open("transport-details.txt", ios::app);

fout << trip\_id << " " << transport\_id << " "<< transport\_type << " " << transport\_cost << endl;

fout.close();

}

void cancelTransport(int delete\_trip\_id){

ifstream fin;

string readLine;

ofstream fout;

fin.open("transport-details.txt", ios::in);

if(fin.is\_open())

{

while(getline(fin, readLine))

{

int fileTripId = 0;

int size = readLine.length();

char tripD[size+1];

int temp = 0, i=0;

strcpy(tripD, readLine.c\_str());

while((temp = tripD[i++]) != 32)

fileTripId = fileTripId \* 10 + (temp - 48);

if(fileTripId != delete\_trip\_id)

{

fout.open("temp.txt", ios::app);

fout<<readLine<<endl;

fout.close();

}

}

fin.close();

}

remove("transport-details.txt");

rename("temp.txt","transport-details.txt");

}

};

class Hotel{

protected:

string hotel\_name;

string check\_in\_date;

string check\_out\_date;

int hotel\_cost;

public:

Hotel() {

hotel\_name = "";

check\_in\_date = "";

check\_out\_date = "";

hotel\_cost = 0;

}

void printHotelMenu(int LINELENGTH){

string border(LINELENGTH,'\*');

cout << endl << border;

cout << "|" << left << setw(20) << "Sr No." << " | " << left << setw(30) << "Hotel Name" << " | " << left << setw(22) << "Price/Day" << right << "|";

cout << border;

cout << "|" << left << setw(20) << " 1" << " | " << left << setw(30) << "ITC Gardenia" << " | " << left << setw(22) << "5000" << right << "|";

cout << "|" << left << setw(20) << " 2" << " | " << left << setw(30) << "Novotel" << " | " << left << setw(22) << "3000" << right << "|";

cout << "|" << left << setw(20) << " 3" << " | " << left << setw(30) << "Woodville Palace" << " | " << left << setw(22) << "2500" << right << "|";

cout << "|" << left << setw(20) << " 4" << " | " << left << setw(30) << "Ibis" << " | " << left << setw(22) << "2000" << right << "|";

cout << border << endl;

}

void setHotelNameCost(int& hotelOption){

SelectHotel:

cout << "\n Enter Hotel sr.no: ";

cin >> hotelOption;

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

switch(hotelOption){

case 1: hotel\_name = "ITC Gardenia";

hotel\_cost = 5000;

break;

case 2: hotel\_name = "Novotel";

hotel\_cost = 3000;

break;

case 3: hotel\_name = "Woodville Palace";

hotel\_cost = 2500;

break;

case 4: hotel\_name = "Ibis";

hotel\_cost = 2000;

break;

default: cout << "\n Enter Correct option from above..." << endl;

goto SelectHotel;

break;

}

}

void getHotelDetails(int LINELENGTH){

int hotelOption = 0;

printHotelMenu(LINELENGTH);

setHotelNameCost(hotelOption);

cout << " Checkin date: ";

getline(cin, check\_in\_date);

cout << " Checkout date: ";

getline(cin, check\_out\_date);

}

void bookHotel(int trip\_id){

ofstream fout;

fout.open("Hotel-details.txt", ios::app);

fout << trip\_id << " " << hotel\_name << " "<< check\_in\_date << " " << check\_out\_date << " " << hotel\_cost << endl;

fout.close();

}

void cancelHotel(int delete\_trip\_id){

ifstream fin;

string readLine;

ofstream fout;

fin.open("Hotel-details.txt", ios::in);

if(fin.is\_open())

{

while(getline(fin, readLine))

{

int fileTripId = 0;

int size = readLine.length();

char tripD[size+1];

int temp = 0, i=0;

strcpy(tripD, readLine.c\_str());

while((temp = tripD[i++]) != 32)

fileTripId = fileTripId \* 10 + (temp - 48);

if(fileTripId != delete\_trip\_id)

{

fout.open("temp.txt", ios::app);

fout<<readLine<<endl;

fout.close();

}

}

fin.close();

}

remove("Hotel-details.txt");

rename("temp.txt","Hotel-details.txt");

}

};

class Traveller {

protected:

string name;

string address;

string contact;

string gender;

public:

Traveller(){

name = "";

address = "";

gender = "";

contact = "";

}

void setTDetails(){

cout << "\n Name: ";

getline(cin, name);

cout << " Gender(M/F): ";

getline(cin, gender);

cout << " Address: ";

getline(cin, address);

cout << " Contact: ";

getline(cin, contact);

}

void storeTDetails(int trip\_id){

ofstream fout;

fout.open("traveller-details.txt", ios::app);

fout << trip\_id << " " << name << " " << gender << " " << address << " " << contact << endl;

fout.close();

}

void deleteTravellerDetails(int delete\_trip\_id){

ifstream fin;

string readLine;

ofstream fout;

fin.open("traveller-details.txt", ios::in);

if(fin.is\_open())

{

while(getline(fin, readLine))

{

int fileTripId = 0;

int size = readLine.length();

char tripD[size+1];

int temp = 0, i=0;

strcpy(tripD, readLine.c\_str());

while((temp = tripD[i++]) != 32)

fileTripId = fileTripId \* 10 + (temp - 48);

if(fileTripId != delete\_trip\_id)

{

fout.open("temp.txt", ios::app);

fout<<readLine<<endl;

fout.close();

}

}

fin.close();

}

remove("traveller-details.txt");

rename("temp.txt","traveller-details.txt");

}

};

class Receipt {

protected:

int receipt\_id;

int sub\_total;

int tax;

int grand\_total;

public:

Receipt() {

receipt\_id = 0;

sub\_total = 0;

tax = 0;

grand\_total = 0;

}

void generateReceipt(int trip\_id, int transport\_cost, int hotel\_cost, int noOfDays, string name){

receipt\_id = generateRandomId();

sub\_total = (hotel\_cost \* noOfDays) + transport\_cost;

tax = (sub\_total \* 18)/100;

grand\_total = sub\_total + tax;

cout << grand\_total << endl;

printReceipt(trip\_id, sub\_total, tax, grand\_total, transport\_cost, hotel\_cost, noOfDays, name);

}

void printReceipt(int trip\_id, int sub\_total, int tax, int grand\_total, int transport\_cost, int hotel\_cost, int noOfDays, string name){

int hotelTotal = sub\_total - transport\_cost;

clearScreen();

string border(LINELENGTH, '^');

cout << border;

print(CENTRE, "Trip Booked", LINELENGTH);

cout << endl << border;

cout << "Receipt -> Loading...";

wait(4000);

clearScreen();

print(LINELENGTH);

cout << border;

print(CENTRE, "Booking Receipt", LINELENGTH);

cout << endl << border;

cout << "Trip Id: " << trip\_id << endl;

cout << "Name: " << name << endl;

cout << "Days(stay): " << noOfDays << endl;

cout << endl << border;

cout << left << setw(20) << "SR.NO" << left << setw(20) << "Description" << right << setw(40) << "Total Price";

cout << endl << border;

cout << left << setw(20) << "01" << left << setw(20) << "Transport" << right << setw(40) << setprecision(2) << transport\_cost;

cout << left << setw(20) << "02" << left << setw(20) << "Hotel" << right << setw(40) << setprecision(2) << hotelTotal;

cout << left << setw(60) << " " << right << setfill('-') << setw(20) << "-";

cout << left << setfill(' ') << setw(40) << " " << left << setw(20) << "SUBTOTAL:" << right << setw(20) << sub\_total;

cout << left << setw(40) << " " << left << setw(20) << "TAX(18%):" << right << setw(20) << tax;

cout << left << setw(40) << " " << left << setw(20) << "GRAND TOTAL:" << right << setw(20) << grand\_total;

cout << endl << border;

}

void storeReceipt(int trip\_id, int transport\_cost, int hotel\_cost, int noOfDays, string name){

generateReceipt(trip\_id, transport\_cost, hotel\_cost, noOfDays, name);

ofstream fout;

fout.open("Receipt-details.txt", ios::app);

fout << trip\_id << " " << receipt\_id << " " << sub\_total << " " << grand\_total << endl;

fout.close();

}

void deleteReceipt(int delete\_trip\_id){

ifstream fin;

string readLine;

ofstream fout;

fin.open("Receipt-details.txt", ios::in);

if(fin.is\_open())

{

while(getline(fin, readLine))

{

int fileTripId = 0;

int size = readLine.length();

char tripD[size+1];

int temp = 0, i=0;

strcpy(tripD, readLine.c\_str());

while((temp = tripD[i++]) != 32)

fileTripId = fileTripId \* 10 + (temp - 48);

if(fileTripId != delete\_trip\_id)

{

fout.open("temp.txt", ios::app);

fout<<readLine<<endl;

fout.close();

}

}

fin.close();

}

remove("Receipt-details.txt");

rename("temp.txt","Receipt-details.txt");

}

};

class Trip:public Traveller, Transport, Hotel, Receipt{

protected:

int trip\_id;

string trip\_start\_date;

string trip\_end\_date;

string trip\_to;

string trip\_from;

int noOfDays;

public:

Trip() {

trip\_id = 0;

trip\_start\_date = "";

trip\_end\_date = "";

trip\_to = "";

trip\_from = "";

noOfDays = 0;

}

void printTripMenu(int LINELENGTH){

string border(LINELENGTH,'\*');

cout << endl << border;

cout << "|" << left << setw(20) << " Trip option" << " | " << left << setw(55) << "Trip route" << right << "|";

cout << border;

cout << "|" << left << setw(20) << " 1" << " | " << left << setw(55) << "Goa to Bangalore" << right << "|";

cout << "|" << left << setw(20) << " 2" << " | " << left << setw(55) << "Goa to Mumbai" << right << "|";

cout << "|" << left << setw(20) << " 3" << " | " << left << setw(55) << "Goa to Delhi" << right << "|";

cout << "|" << left << setw(20) << " 4" << " | " << left << setw(55) << "Goa to Shimla" << right << "|";

cout << border << endl;

}

void setTripDestinations(int& tripOption){

SelectTrip: cout << " Enter trip option: ";

cin >> tripOption;

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

switch(tripOption){

case 1: trip\_from = "Goa";

trip\_to = "Bengaluru";

break;

case 2: trip\_from = "Goa";

trip\_to = "Mumbai";

break;

case 3: trip\_from = "Goa";

trip\_to = "Delhi";

break;

case 4: trip\_from = "Goa";

trip\_to = "Shimla";

break;

default: cout << "\n Enter Correct option from above..." << endl;

goto SelectTrip;

break;

}

}

void getTripDetails(int LINELENGTH){

int tripOption = 0;

trip\_id = generateRandomId();

// set traveller details

print(CENTRE,"Traveller Details", LINELENGTH);

setTDetails();

cout << endl;

// set trip details

print(CENTRE,"Trip Details", LINELENGTH);

printTripMenu(LINELENGTH);

setTripDestinations(tripOption);

cout << " Trip start date: ";

getline(cin, trip\_start\_date);

cout << " Trip end date: ";

getline(cin, trip\_end\_date);

cout << " Days(stay): ";

cin >> noOfDays;

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

cout << endl;

//set transport details

print(CENTRE,"Transport Details", LINELENGTH);

getTransportDetails(tripOption);

cout << endl;

// set hotel details

print(CENTRE,"Hotel Details", LINELENGTH);

getHotelDetails(LINELENGTH);

}

void storeTripDetails(){

ofstream fout;

fout.open("trip-details.txt", ios::app);

fout << trip\_id << " " << trip\_start\_date << " " << trip\_end\_date << " " << trip\_from << " " << trip\_to << " " << noOfDays << endl;

fout.close();

}

void bookTrip(int LINELENGTH){

clearScreen();

print(LINELENGTH); // prints the header

getTripDetails(LINELENGTH);

storeTDetails(trip\_id);

bookTransport(trip\_id);

bookHotel(trip\_id);

storeTripDetails();

storeReceipt(trip\_id, transport\_cost, hotel\_cost, noOfDays, name);

}

void deleteTripDetails(int delete\_trip\_id){

clearScreen();

print(LINELENGTH);

ifstream fin;

string readLine;

ofstream fout;

fin.open("trip-details.txt", ios::in);

if(fin.is\_open())

{

while(getline(fin, readLine))

{

int fileTripId = 0;

int size = readLine.length();

char tripD[size+1];

int temp = 0, i=0;

strcpy(tripD, readLine.c\_str());

while((temp = tripD[i++]) != 32)

fileTripId = fileTripId \* 10 + (temp - 48);

if(fileTripId != delete\_trip\_id)

{

fout.open("temp.txt", ios::app);

fout<<readLine<<endl;

fout.close();

}

}

fin.close();

}

remove("trip-details.txt");

rename("temp.txt","trip-details.txt");

}

void cancelTrip(){

string border(LINELENGTH, '^');

int delete\_trip\_id = 0;

cout << "Enter trip id of the trip to be cancelled: ";

cin >> delete\_trip\_id;

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

deleteTripDetails(delete\_trip\_id);

cancelTransport(delete\_trip\_id);

cancelHotel(delete\_trip\_id);

deleteTravellerDetails(delete\_trip\_id);

deleteReceipt(delete\_trip\_id);

cout << border;

print(CENTRE, "Trip Cancelled", LINELENGTH);

cout << endl << border;

}

};

int main()

{

clearScreen();

system("title Travel Agency System");

// Variables

int chooseOneFromMenu = 0;

char exitSurity;

Trip tripObject;

// menu

MainMenu:

clearScreen();

print(LINELENGTH);

printMainMenu(LINELENGTH);

cin >> chooseOneFromMenu;

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

switch (chooseOneFromMenu)

{

case 1:

tripObject.bookTrip(LINELENGTH);

break;

case 2:

tripObject.cancelTrip();

break;

case 3:

ExitProgram:

cout << "Program terminating. Are you sure? (Y/N): ";

cin >> exitSurity;

if (exitSurity == 'y' || exitSurity == 'Y') {

return 0;

}else if (exitSurity == 'n' || exitSurity == 'N') {

clearScreen();

goto MainMenu;

}else {

cout << "Next time choose after read the corresponding line." << endl;

goto ExitProgram;

}

break;

default:

cout << "Please choose between 1 - 3.";

clearScreen();

system("PAUSE");

goto MainMenu;

break;

}

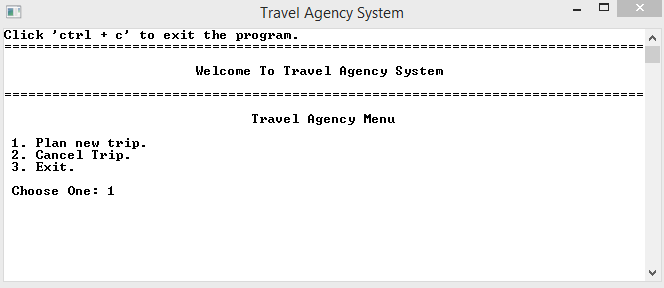
system("PAUSE");

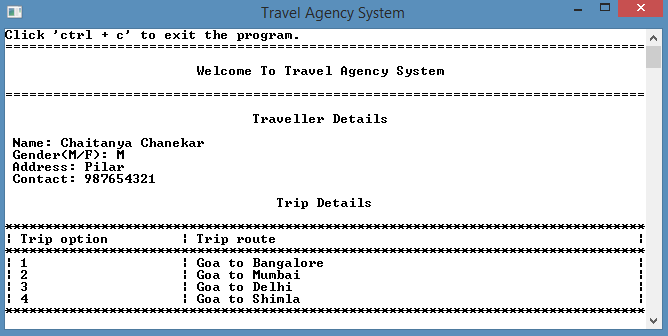
goto MainMenu;

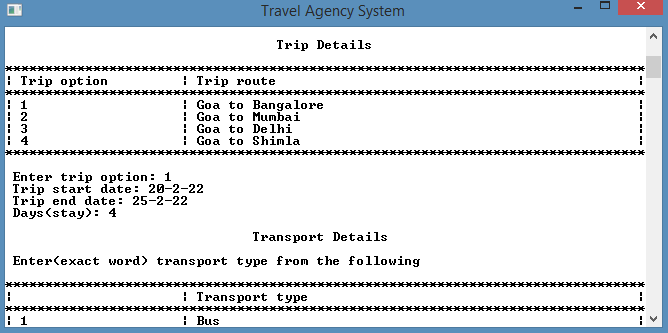
return 0;

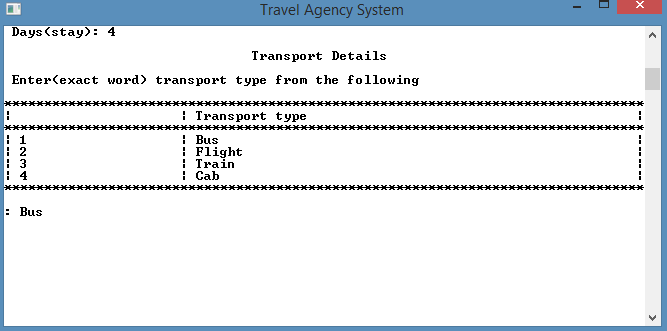
}

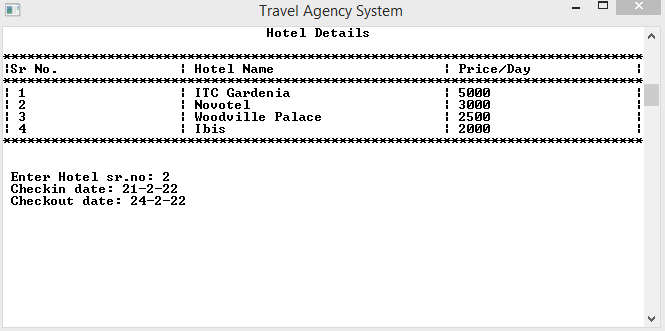
**Output:**

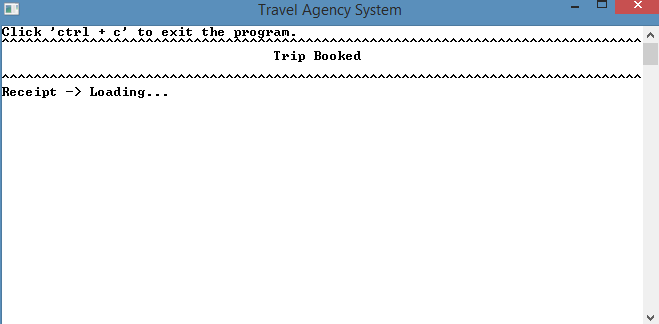




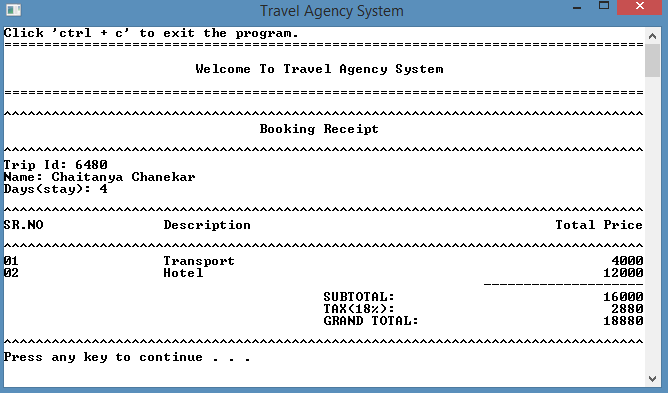




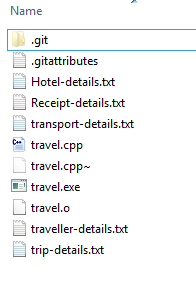
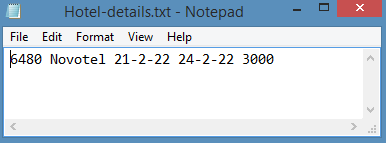


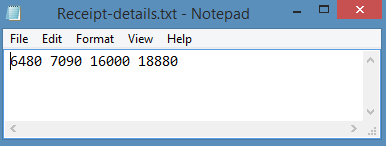


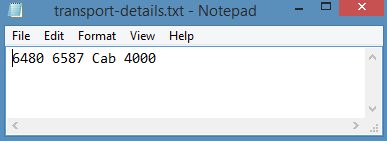
Receipt:

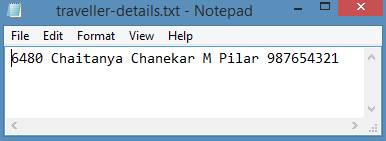


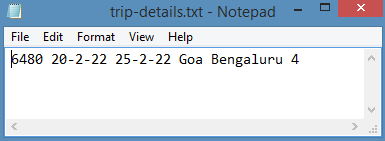
**File system: after booking**



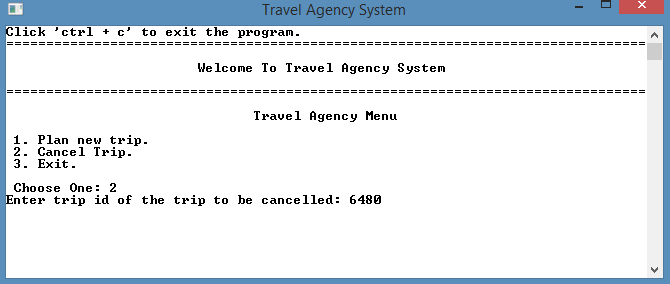


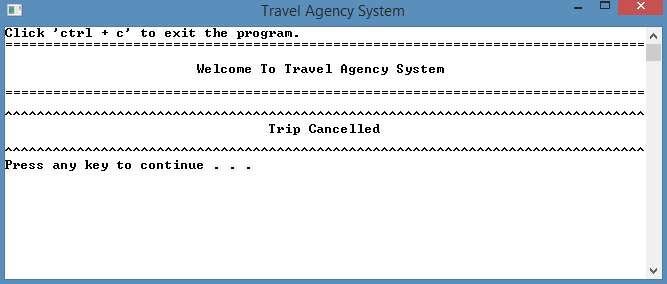






Canceling Trip:





File system after cancelling trip:

