

EECS 560 Lab 7 – Min Heap and Max Heap

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

200 Points

Due date:

11:59 pm, Monday, 04/06/2020 for Tuesday labs

11:59 pm, Wednesday, 04/08/2020 for Thursday labs

Purpose:

The purpose of this lab is to implement the Min 3-Heap and Max 3-Heap data structures in C++.

General Requirements: In this lab, you are required to implement a Patient and Doctor management system using Min 3-Heap and Max 3-Heap data structures. This is to manage patients' treatment order and doctors' assignment order. Each requirement will be handled separately by the application.

Technical Specifications: Both Min and Max Heaps should be implemented using array-based implementations.

In Min 3-Heap (and Max 3-Heap):

- The root of T is at $A[0]$.
- The parent of $A[i]$ is at $A[\text{floor}((i-1)/3)]$, if it exists.
- The j^{th} child of $A[i]$ is at $A[3i+j]$, $1 \leq j \leq 3$, if it exists.

Use the **Bottom-up method** for building the heap. (If a different approach is used, the results might be different.) For the Bottom-up approach, make sure that:

- After insertion/deletion, you maintain a complete 3-tree structure for the heap.
- You re-structure the complete 3-tree from bottom to top. So that it will satisfy the respective heap-ordered tree property.
- You follow an array-based implementation for greater efficiency.

Bottom-up Method: Let's say the details of input data $S = \{5, 3, 2, 6, 8, 5, 1\}$. The Bottom-up method inserts all the values in S into a Min 3- (or Max 3-Heap) structure as in the given order. Then each non-leaf node is scanned from leaf-to-root and from right-to-left). Each node is then compared with its children and the heapify operation is performed if required.

EECS 560 Lab 7 – Min Heap and Max Heap

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

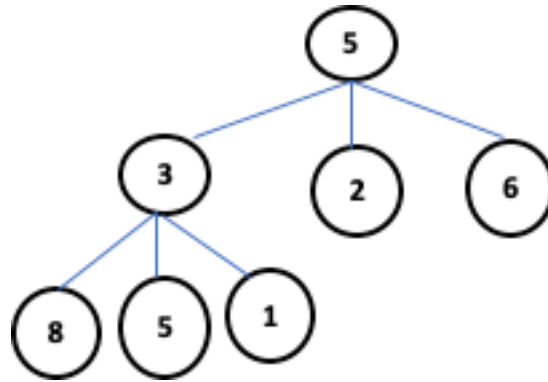


Fig 1. Initial insertion for the Bottom-Up Approach

Now, apply the Bottom-up approach. The first non-leaf node is 3; it should be compared with its children as follows:

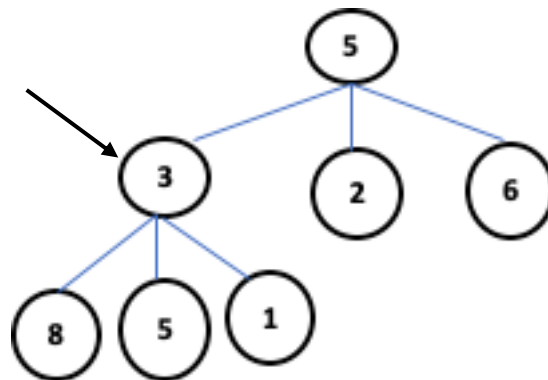


Fig 2. First non-leaf node 3 is to be compared with its descendants

Since this is a Min heap and 3 is greater than 1, we need to swap the 3 with the 1.

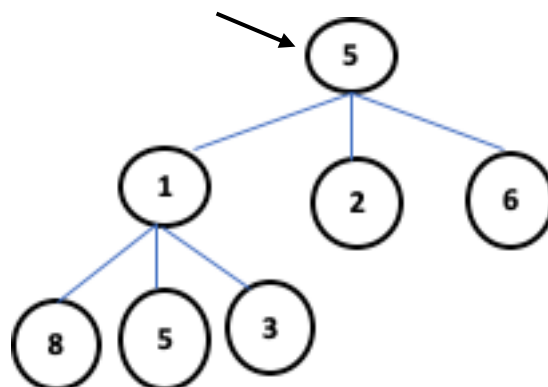


Fig 3. Next non-leaf node 5 is to be compared with its descendants

The next non-leaf node is 5, which is the root. Compare the 5 to its children. 5 is greater than 1. Swap 5 with 1.

EECS 560 Lab 7 – Min Heap and Max Heap

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

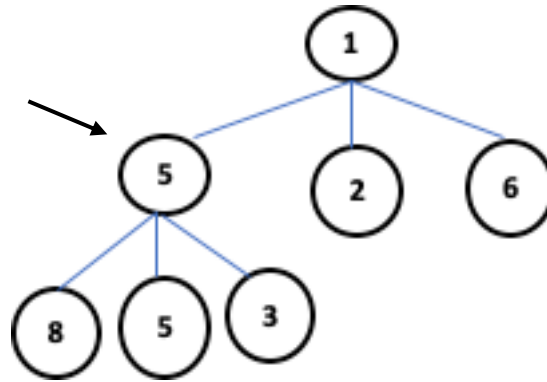


Fig 4. Node 5 to be swapped with 3

Compare 5 with its children. 5 is greater than 3. Swap 5 with 3 in above figure to maintain the heap property.

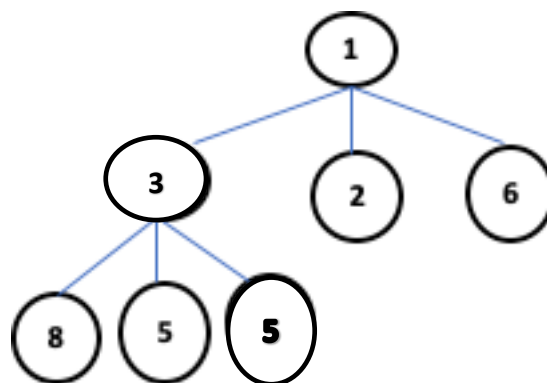


Fig 5. Final Min Heap Structure

The above is the final heap structure using the Bottom Up approach of Min Heap.

Lab Data Specifications: There will be 2 input data files named doctors.txt and patients.txt which hold the data for the doctors and the patients simultaneously. Data from the input files will be stored in priority order using a heap structure. Below are the details for the input files:

doctors.txt: This file should contain the below values:

<Doctor's first name><Doctor's last name><Patients assigned>

Doctor's first name: First name of the doctor. A string value of lower case.

Doctor's last name: Last name of the doctor. A string value of lower case.

Patients assigned: Number of patients assigned to the doctor. Can be zero for a newly added doctor. However, each doctor has a limit of 25 patients assigned to him/her.

EECS 560 Lab 7 – Min Heap and Max Heap

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

Sample data: doctors.txt

jake,co,5
steve,bl,7
dave,ck,11
neon,bh,25
tomjJason,9
rashid,peny,25
austin,eker,25
peyon,barber12
john,kedy,2
tim,gy,4

patients.txt: This file should contain the below values:

<Patient's first name><Patient's last name>< Patient's urgency>

Patient's first name: First name of the patient. A string value of lower case.

Patient's last name: Last name of the patient. A string value of lower case.

Patient's urgency: Importance value of the patient's issue. If the value is high, then the patient needs to be treated with high priority.

Sample data: patients.txt

adam,conner,22
peyton,barber,1
john,kenny,23
tevin,coleman,10
royce,freeman,2
derrius,guice,15
chris,thompson,14
dan,harris,21
damien,harris,5
duke,johnson,9
rashaad,penny,28
austin,ekeler,4

Application Specifications: Based on the input files provided, the data needs to be inserted into the corresponding Max or Min heap structure. **Patients are prioritized based on their urgency, and doctors are prioritized based on the number of patients assigned.** As the execution starts, the program should insert doctor and patient data into the appropriate heap structure and continue to show the below screen of options.

EECS 560 Lab 7 – Min Heap and Max Heap

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

-----Hospital Patient & Doctor Management System-----

1. Patient Management
2. Doctor Assignment
3. Exit

Patient Management: The list of patients is managed in this section. As you execute the code, the details from the patients.txt file should be inserted into the appropriate heap. **Each patient from the input file will be added to the heap based on the urgency value, with the highest urgency having the priority to be treated first.** Patient urgency can be any integer. Then once the user chooses to go to patient management, the program should show the below options.

-----Patient Management System-----

1. New Patient
2. Treat Patient
3. Change Urgency
4. Next Patient
5. Last Patient
6. Patient Token
7. Patient Count
8. Patient Report
9. Go Back to Menu

New Patient(): Add a new patient to the system. A new record of the patient with details <first name><last name><urgency> is to be added to the heap structure. The user should be prompted to enter the required data for the patient, and then the patient record should be added. All the records need to be inserted into the heap according to the patient's priority. At every insertion, the program should Heapify to re-structure all of the records according to the corresponding data structure's priority. **Duplicate records are not allowed.**

Treat Patient(): Remove a patient from the records. Always remove a patient record with high priority. Technically, remove a high priority record from the heap structure. **The user will not be prompted with anything.** Once the user chooses this option, the highest priority patient should be treated. Print the details of the patient while removing him/her from the heap.

Change Urgency(): Change the urgency level. For the patient record, the corresponding urgency level can be increased/decreased. The user should be prompted to enter the patient details, i.e., the first name and last name. Based on this, find the patient record. Print the details of patient, and prompt the user to enter a new urgency value. Technically, based on the new urgency value, the record should be re-Inserted into the heap structure. That is, remove the existing record and insert with a new urgency value. **It is required to Heapify again for this option.**

EECS 560 Lab 7 – Min Heap and Max Heap

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

Next Patient(): Print the next patient to be treated. Technically, print the details for the patient with highest priority in the heap structure. Print the patient name (first and last names) and urgency.

Last Patient(): Print the last patient to be treated, that is the patient with the lowest priority. Technically, print the details for the patient with lowest priority in the heap structure. The patient name (first and last names) and urgency are to be printed.

Patient Token(): This is to find the position of the patient in the queue. Technically, all patient records are inserted based on the priorities. Here, the user should be prompted to enter the patient details and find when (at what number priority) that patient is treated. For example: if the patient has the 3rd priority in the queue, then the token for that patient is 3. The patient with highest priority will have the token 1.

Patients Count(): Print the count for the number of patients. Technically, print the count of elements in the Heap structure.

Patients Report(): Print all the patient details (first name, last name, urgency) In level order for the heap. Technically, print all of the records for the heap in level order.

Go Back to Menu(): The program should go back to the main menu for the Hospital Patient & Doctor Management System and should not exit here. In particular, the **program should return to the main menu and not exit from the program execution.**

Doctor Assignment: The doctor's count of patient assignments is handled in this section. As you execute the code, the details from the doctors.txt file should be inserted into the appropriate heap. Each doctor from the input file will be added to the heap based on the patient count. **The doctor with the lowest count will have the priority to be assigned a new patient first.** The patient count for a doctor should not be more than 25. This limit is explained in the options below.

----- Doctor Assignment System-----

1. New Doctor
2. Update Patients
3. Remove Doctor
4. Next Available Doctor
5. Check Availability
6. Total Doctor Count
7. Available Doctor Count
8. Available Doctors Report
9. Busy Doctors List
10. Go Back to Menu

New Doctor(): Add a new doctor to the system. A new record of the doctor with details <first name><last name><number of patients> is to be added to heap structure. **Note that a new doctor will always have 0 patients assigned.** The user should be prompted to enter the

EECS 560 Lab 7 – Min Heap and Max Heap

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

required data for the doctor, and then the doctor record should be added. All the records need to be inserted according to the priority in the heap. For every insertion, the program should Heapify all the records in the structure according to the corresponding data structure's priority. **Duplicate doctor names are not allowed, but duplicate patient counts are allowed.**

Update Patients(): Change the count of the patients for a doctor. For a doctor's record, the corresponding patient's count can be increased/decreased. The user should be prompted to enter the doctor details, i.e., the first name and last name. Based on this, find the doctor's record. Print the details of the doctor, and prompt the user if the patient count needs to be increased/reduced. If increase is selected, increase the patient count as per the user input. If reduce is selected, decrease the patient count as per the user input. Technically, based on the updated patient count within the doctor's record, the records should be re-inserted into the Heap structure. That is, remove the existing record and insert the records with updated patient counts. **It is required to Heapify again for this option.**

Remove Doctor(): Remove a doctor from the records. Always remove the doctor record with the highest priority. Technically, remove a record with highest priority from the heap structure. **The user will not be prompted with anything.** Once the user chooses this option, the highest priority doctor (that is the one with the lowest patient count) should be deleted. Print the details of the doctor while removing him/her from the heap.

Next Available Doctor(): Print the next available doctor in the list. Technically, print the details for the doctor with highest priority in the heap structure. The doctor's name (i.e., first and last names) and patient count are to be printed.

Check Availability(): Check whether or not a doctor is currently available. The user should be prompted to enter the doctor details (i.e., the first name and last name). Then based on the name, retrieve the availability of the doctor. The doctor is available if the patient count is < 22 and not available if the patient count is ≥ 22 .

Total Doctor Count(): Print the count for the number of doctors. Technically, print the number of elements in the heap structure.

Available Doctor Count(): Print the count for the number of doctors that are available. Technically, print the number of elements in the heap structure with the patient count < 22 .

Available Doctors Report(): Print the doctor details (first name, last name, patient count) In level order of the heap. Technically, print all the records with patient count < 22 in the heap according to level order.

Busy Doctors List(): Print the doctor details (first name, last name, patient count) for the doctors that are not available. Technically, print all the records for the patients with count ≥ 22 in the heap.

Go Back to Menu(): The program should go back to the main menu of the Hospital Patient & Doctor Management System and should not exit here. In particular, the **program should return to the main menu and not exit from the program execution.**

EECS 560 Lab 7 – Min Heap and Max Heap

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

Expected Results:

Please note that you are not expected to show the heap graphically in your output. **However, your output should show the results of both the doctors and patients simultaneously.** Below are the expected results of performing the various operations. **The following outputs are just for illustrative purposes only.** As you execute the code, your program should build both heaps according to the input data and show the below options.

-----Hospital Patient & Doctor Management System-----

1. Patient Management
2. Doctor Assignment
3. Exit

Please choose one option:

> 1

-----Patient Management System-----

1. New Patient
2. Treat Patient
3. Change Urgency
4. Next Patient
5. Last Patient
6. Patient Token
7. Patients Count
8. Patients Report
9. Go Back to Menu

Please choose your option:

>1

>Enter Patient's First name:

> smith

>Enter Patient's Last name:

>ito

>Enter Patient's Urgency:

>27

> New Patient record added

-----Patient Management System-----

1. New Patient
2. Treat Patient
3. Change Urgency
4. Next Patient
5. Last Patient

EECS 560 Lab 7 – Min Heap and Max Heap

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

6. Patient Token
7. Patients Count
8. Patients Report
9. Go Back to Menu

Please choose your option:

>8

> Current Patients report:

rashaad,penny,28

adam,conne,22

derrius,guic,15

ito,smith,27

peyton,barber,1

royce,freeman,2

chris,thompson,14

dan,harris,21

damien,harris,5

duke,johnson,9

john,kenny,23

tevin,coleman,10

austin,ekeler,4

-----Patient Management System-----

1. New Patient
2. Treat Patient
3. Change Urgency
4. Next Patient
5. Last Patient
6. Patient Token
7. Patients Count
8. Patients Report
9. Go Back to Menu

Please choose your option:

>7

>Current patient count is: 12

-----Patient Management System-----

1. New Patient
2. Treat Patient
3. Change Urgency
4. Next Patient
5. Last Patient
6. Patient Token
7. Patients Count

EECS 560 Lab 7 – Min Heap and Max Heap

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

8. Patients Report
9. Go Back to Menu

Please choose your option:

>6

> Enter details of patient:

>First name: john

>Last name: kenny

>Patient token: 3

-----Patient Management System-----

1. New Patient
2. Treat Patient
3. Change Urgency
4. Next Patient
5. Last Patient
6. Patient Token
7. Patients Count
8. Patients Report
9. Go Back to Menu

Please choose your option:

>5

> Last patient details: peyton,barber,1

-----Patient Management System-----

1. New Patient
2. Treat Patient
3. Change Urgency
4. Next Patient
5. Last Patient
6. Patient Token
7. Patients Count
8. Patients Report
9. Go Back to Menu

Please choose your option:

>4

>Next patient: rashaad,penny,28

EECS 560 Lab 7 – Min Heap and Max Heap

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

-----Patient Management System-----

1. New Patient
2. Treat Patient
3. Change Urgency
4. Next Patient
5. Last Patient
6. Patient Token
7. Patients Count
8. Patients Report
9. Go Back to Menu

Please choose your option:

>3

> Enter the patient details to change the urgency:

>First name: john

>Last name: kenny

> Current details of this patient are: john,kenny,23

> Enter new urgency level: 29

> The patient record is updated.

-----Patient Management System-----

1. New Patient
2. Treat Patient
3. Change Urgency
4. Next Patient
5. Last Patient
6. Patient Token
7. Patients Count
8. Patients Report
9. Go Back to Menu

Please choose your option:

>2

> High priority patient successfully treated. Below are the details:

> john,kenny,29

-----Patient Management System-----

1. New Patient
2. Treat Patient
3. Change Urgency
4. Next Patient
5. Last Patient
6. Patient Token
7. Patients Count

EECS 560 Lab 7 – Min Heap and Max Heap

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

8. Patients Report
9. Go Back to Menu

Please choose your option:

>9

-----Hospital Patient & Doctor Management System-----

-

1. Patient Management
2. Doctor Assignment
3. Exit

Please choose your option:

>2

----- Doctor Assignment System-----

1. New Doctor
2. Update Patients
3. Remove Doctor
4. Next Available Doctor
5. Check Availability
6. Total Doctor Count
7. Available Doctor Count
8. Available Doctors Report
9. Busy Doctors List
10. Go Back to Menu

Please choose your option:

>1

> Enter doctor details:

> First name: joe

> Last name: mixon

> A new doctor is added and available.

----- Doctor Assignment System-----

1. New Doctor
2. Update Patients
3. Remove Doctor
4. Next Available Doctor
5. Check Availability
6. Total Doctor Count
7. Available Doctor Count
8. Available Doctors Report
9. Busy Doctors List

EECS 560 Lab 7 – Min Heap and Max Heap

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

10. Go Back to Menu

Please choose your option:

>10

> Doctors not available at this time are:

neon,bh,25

rashaad,penny,25

austin,ekeler,25

----- Doctor Assignment System-----

1. New Doctor
2. Update Patients
3. Remove Doctor
4. Next Available Doctor
5. Check Availability
6. Total Doctor Count
7. Available Doctor Count
8. Available Doctors Report
9. Busy Doctors List
10. Go Back to Menu

Please choose your option:

>9

The available doctors are:

joe,mixon,0

steve,bl,7

tim,gy,4

john,kedy,2

tom,jason,9

dave,ck,11

jake,co,5

----- Doctor Assignment System-----

1. New Doctor
2. Update Patients
3. Remove Doctor
4. Next Available Doctor
5. Check Availability
6. Total Doctor Count
7. Available Doctor Count
8. Available Doctors Report
9. Busy Doctors List
10. Go Back to Menu

Please choose your option:

EECS 560 Lab 7 – Min Heap and Max Heap

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

>8

> There are 7 doctors available.

----- Doctor Assignment System-----

1. New Doctor
2. Update Patients
3. Remove Doctor
4. Next Available Doctor
5. Check Availability
6. Total Doctor Count
7. Available Doctor Count
8. Available Doctors Report
9. Busy Doctors List
10. Go Back to Menu

Please choose your option:

>7

> There are 11 total doctors.

----- Doctor Assignment System-----

1. New Doctor
2. Update Patients
3. Remove Doctor
4. Next Available Doctor
5. Check Availability
6. Total Doctor Count
7. Available Doctor Count
8. Available Doctors Report
9. Busy Doctors List
10. Go Back to Menu

Please choose your option:

>6

> Enter doctor details to check availability:

>First name: tom

>Last name: jason

> This doctor is available.

EECS 560 Lab 7 – Min Heap and Max Heap

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

----- Doctor Assignment System-----

1. New Doctor
2. Update Patients
3. Remove Doctor
4. Next Available Doctor
5. Check Availability
6. Total Doctor Count
7. Available Doctor Count
8. Available Doctors Report
9. Busy Doctors List
10. Go Back to Menu

Please choose your option:

>5

>Next Doctor available: joe mixon

----- Doctor Assignment System-----

1. New Doctor
2. Update Patients
3. Remove Doctor
4. Next Available Doctor
5. Check Availability
6. Total Doctor Count
7. Available Doctor Count
8. Available Doctors Report
9. Busy Doctors List
10. Go Back to Menu

Please choose your option:

>4

>The doctor with the lowest patient count has been removed.

----- Doctor Assignment System-----

1. New Doctor
2. Update Patients
3. Remove Doctor
4. Next Available Doctor
5. Check Availability
6. Total Doctor Count
7. Available Doctor Count
8. Available Doctors Report
9. Busy Doctors List
10. Go Back to Menu

EECS 560 Lab 7 – Min Heap and Max Heap

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

Please choose your option:

>3

> Enter the doctor's details to update the patient count:

>First name: tom

>Last name: jason

> Current patient count: 15

> Select 1 to add patients or 0 to reduce patients: 0

>Enter the updated count: 10

> The doctor's record is updated.

----- Doctor Assignment System-----

1. New Doctor
2. Update Patients
3. Remove Doctor
4. Next Available Doctor
5. Check Availability
6. Total Doctor Count
7. Available Doctor Count
8. Available Doctors Report
9. Busy Doctors List
10. Go Back to Menu

Please choose your option:

>11

-----Hospital Patient & Doctor Management System-----

-

1. Patient Management
2. Doctor Assignment
3. Exit

Please choose your option:

>3

> Bye Bye!

Report: Answer the following questions. Your answers should not exceed 5 to 6 sentences.

1. Explain why it is advantageous to use priority heaps in hospital management systems.
2. Explain a few features which should be incorporated into the hospital patient and doctor management system for the system to be useful in an actual (i.e., real-world hospital) for managing emergency room patient cases.
3. If we need to apply priority heaps in product recommendations (based on features or product ratings), how can we apply them?

EECS 560 Lab 7 – Min Heap and Max Heap

Prof.: Dr. Shontz, GTAs: Chiranjeevi Pippalla, Prashanthi Mallojula

Grading rubric:

- Full grade: The program should execute without any issues with all the options executed and with no memory leaks.
- Points will be taken off for execution errors, such as memory leaks, segmentation/program abort issues and missing handling of invalid cases.
- Programs that are compiled but do not execute will earn in the range of 0 to 50% of the possible points. Your grade will be determined based on the program design and the options implemented in the code.

Submission instructions:

- All files, i.e., the source files and Makefile, should be zipped in a folder.
- Include a ReadMe.txt if your code requires any special instructions to run.
- The naming convention of the folder should be LastName_Lab7.zip (or .tar or .rar or .gz).
- Email it to your respective lab instructor: chiranjeevi.pippalla@ku.edu (Chiru) or prashanthi.mallojula@ku.edu (Prashanthi) with subject line EECS 560 Lab7.
- Your program should compile and run on the **Linux machines** in **Eaton 1005D** using **g++**.