



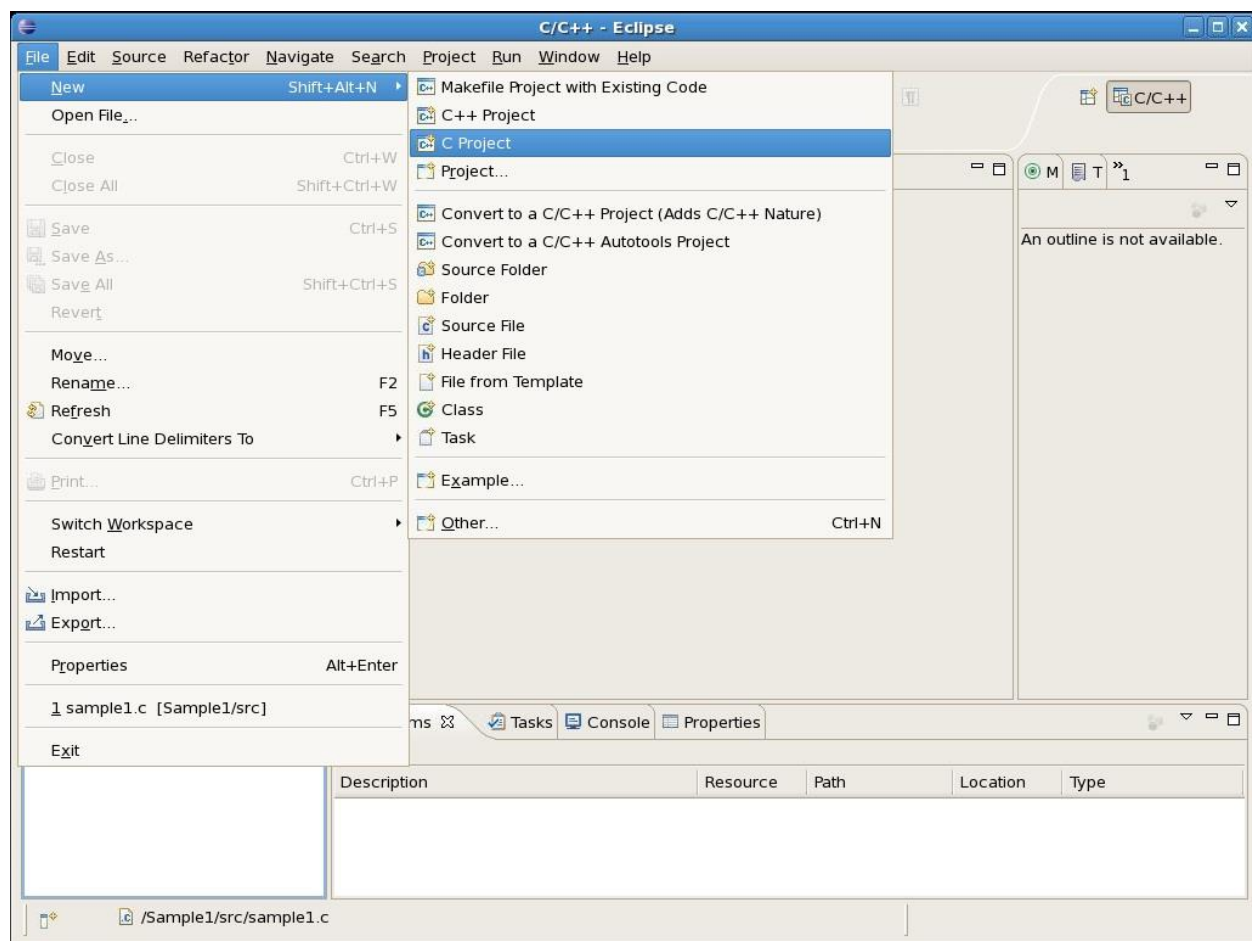
## ΕΡΓΑΣΤΗΡΙΟ 4

### Μέρος Α': Eclipse CDT και Debugging

Στο μάθημα ΕΠΛ131 είδαμε πως μπορούμε να κάνουμε αποσφαλμάτωση ενός προγράμματος Java από το περιβάλλον ανάπτυξης εφαρμογών Eclipse. Το Eclipse υποστηρίζει επίσης και ανάπτυξη προγράμματος σε άλλες γλώσσες προγραμματισμού όπως C/C++, μέσω ενός εργαλείου ανάπτυξης, το CDT (C/C++ Development Tools). Το CDT εκτός από το να προσφέρει ένα περιβάλλον ανάπτυξης για προγράμματα C/C++, επεκτείνει το κανονικό *view Eclipse Debug* με λειτουργίες για τον εντοπισμό σφαλμάτων σε κώδικα C/C++. Για να γίνει αυτό, το CDT χρησιμοποιεί το GNU GDB για τον εντοπισμού σφαλμάτων, το οποίο πρέπει να είναι διαθέσιμος σε τοπικό επίπεδο.

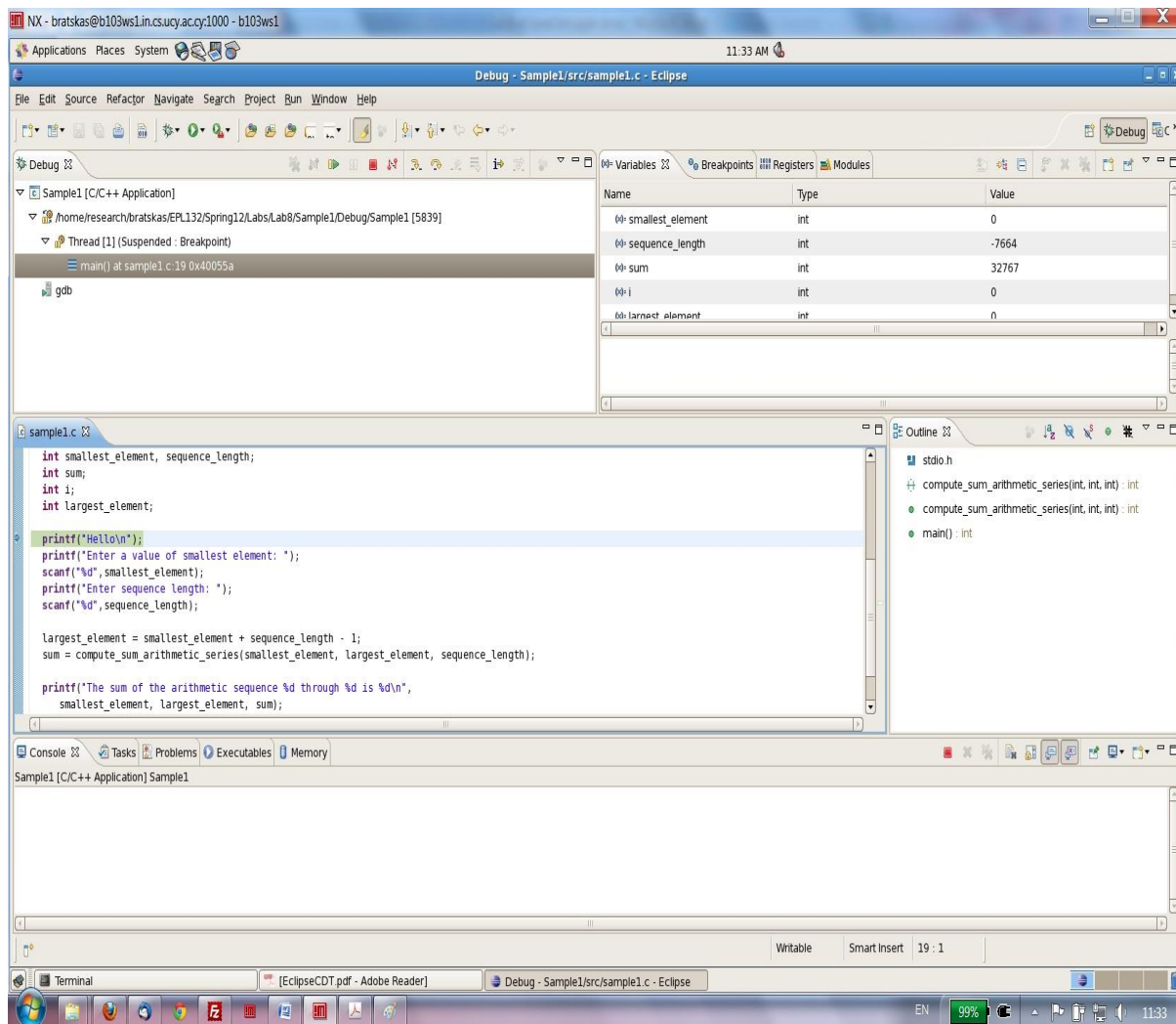
### Debugging C programs in Eclipse CDT

1. Στην ιστοσελίδα του μαθήματος κατεβάστε τα δυο αρχεία `sample1.c` και `sample2.c`.
2. Δημιουργήστε ένα καινούργιο C Project.

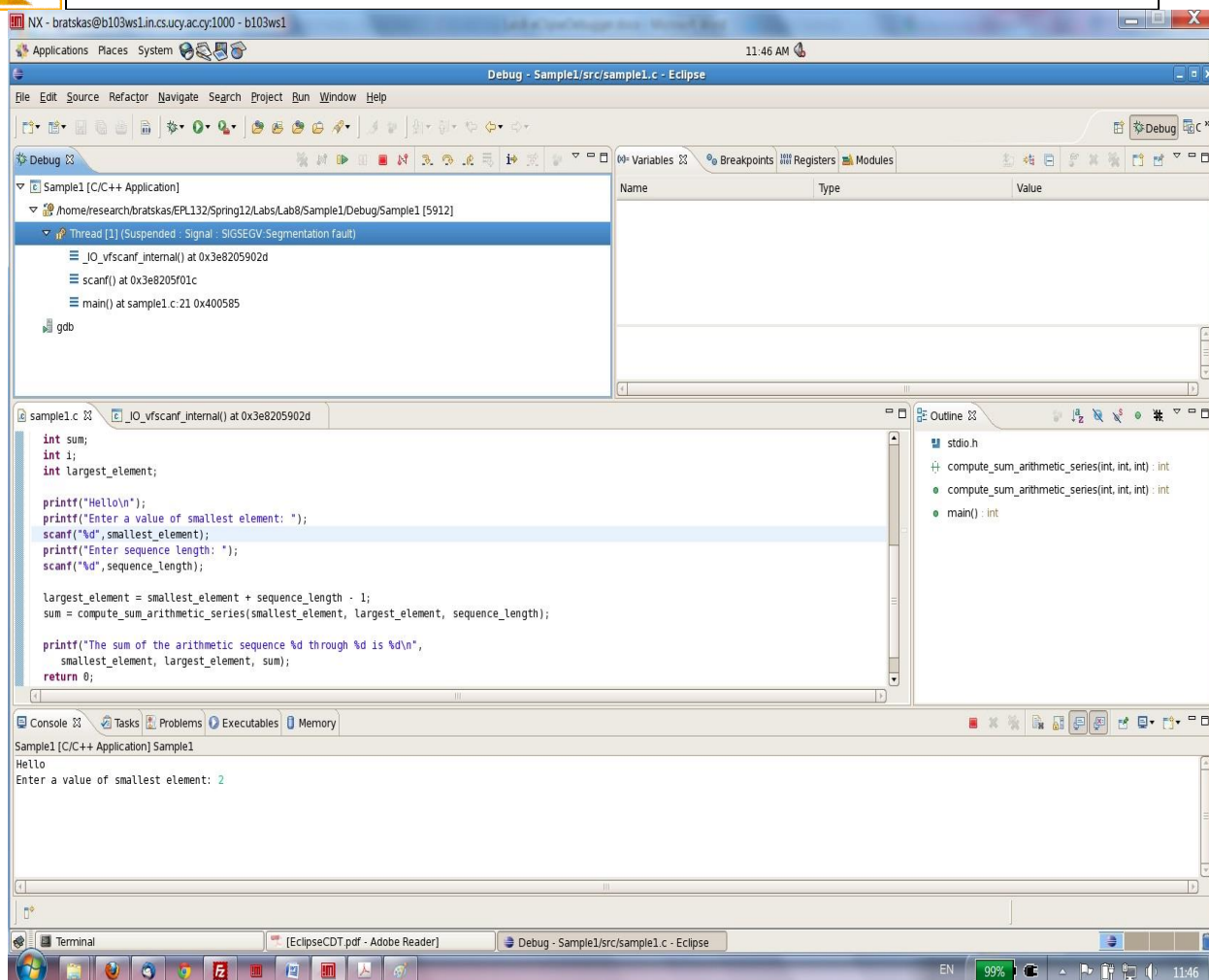




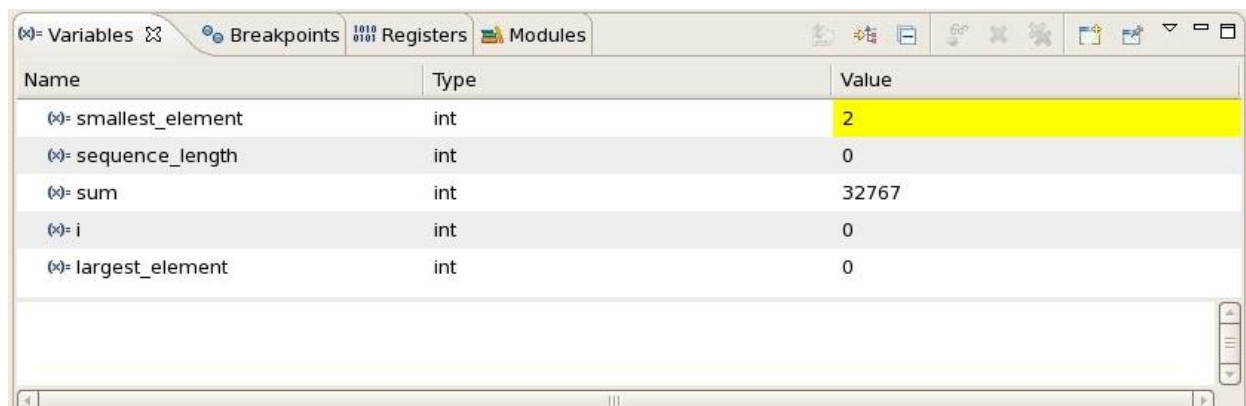
3. Διαλέξτε **Executable -> Empty Project**, στα **Toolchains -> Linux GCC** δώστε ένα όνομα και μετά **Finish**.
4. Έχει δημιουργηθεί ένα κενό project. Προσθέστε σε αυτό το αρχείο `sample1.c` μέσω του **File > Import > General > File System**
5. Επιλέξτε **Project > Build Project** για να μεταγλωττίσετε το πρόγραμμα
6. Προσθέστε ένα **breakpoint** (*double click αριστερά της γραμμής*) στην πρώτη γραμμή της συνάρτησης `main()`
7. Επιλέξτε **Run > Debug**. Το view του Eclipse πρέπει να αλλάξει όπως φαίνεται στην παρακάτω εικόνα.



8. Συνεχίστε την εκτέλεση του προγράμματος βήμα-βήμα πατώντας **F5 (Step Into)**. Στην περίπτωση που η ροή εκτέλεσης φτάσει στο κάλεσμα μιας συνάρτησης, πατώντας **F5** η εκτέλεση θα συνεχιστεί μέσα στο κώδικα της συνάρτησης. Εάν θέλετε να προσπεράσετε αυτό, πρέπει να πατήσετε **F6 (Step Over)**.
9. Όταν ο χρήστης εισάγει έναν αριθμό, το πρόγραμμα βγάζει **Segmentation fault**, όπως στην παρακάτω εικόνα.



10. Το Eclipse σας δείχνει την γραμμή που υπάρχει λάθος στον κώδικα, κάτι που εσείς πρέπει να διορθώσετε.
11. Μετά την διόρθωση, κάντε **Save**, μετά **Build Project** και συνεχίστε την εκτέλεση του προγράμματος, διορθώνοντας και άλλα λάθη εάν υπάρχουν.
12. Κατά την διάρκεια της εκτέλεσης, δείτε στο παράθυρο πάνω δεξιά (**Variable view**), τις αλλαγές στις τιμές των μεταβλητών.





13. Το **Variable view** είναι πολύ χρήσιμο:

- Μπορείτε να αλλάξετε την τιμή μιας μεταβλητής εισάγοντας μια καινούργια τιμή κάτω από το Value.
- Οι τοπικές μεταβλητές εμφανίζονται ανάλογα με την εμβέλεια τούς
- Οι καθολικές μεταβλητές δεν εμφανίζονται, αλλά μπορείτε να το αλλάξετε αυτό κάνοντας right-click στο view και επιλέγοντας **Add Global Variables**
- ένας σύνδεσμος για μια μεταβλητή τύπου δείκτη μπορεί να ακολουθείται, επισημαίνοντας το τρίγωνο δίπλα στο μεταβλητό προς τα κάτω. Αυτό μπορεί να γίνει κατ'επανάληψη, για να εμφανίσετε τα στοιχεία μιας συνδεδεμένης λίστας κλπ.
- αν ένας δείκτης δείχνει σε ένα πίνακα που έχει δημιουργηθεί δυναμικά, ολόκληρος ο πίνακας μπορεί να εμφανίζεται με δεξί κλικ πάνω στην μεταβλητή και επιλέγοντας **Display As Array**
- να επανεκκινήσετε το πρόγραμμα σας από την αρχή, κάντε κλικ στο εικονίδιο **Restart** στο **Debug Toolbar**.

### **Bug hunt!**

- Χρησιμοποιώντας το Eclipse Debugger διορθώστε τα λάθη του προγράμματος parameter.c
- Το πρόγραμμα palindrome.c ελέγχει αν μια συμβολοσειρά είναι παλινδρομική.
  - Μεταγλωττίστε το πρόγραμμα και διορθώστε ό,τι λάθη υπάρχουν.
  - Στο πρόγραμμα υπάρχουν ατέλειες σε μερικές γραμμές του κώδικα.
  - Αλλάξτε το πρόγραμμα έτσι ώστε να υποστηρίζει είσοδο συμβολοσειρών αυθαίρετου μήκους.

### **Μέρος Β': Λειτουργίες σε strings με χρήση pointers**

- Από την ιστοσελίδα του μαθήματος ανοίξτε το αρχείο stringlib.c
  - Υλοποιήστε τις συναρτήσεις που βρίσκονται στο αρχείο (στο αρχείο υπάρχουν σχόλια σχετικά με την λειτουργία τούς)
  - Η συνάρτηση main() σας δίνεται έτοιμη, δεν χρειάζεται να αλλάξετε κάτι.