Σε ένα λειτουργικό σύστημα διατηρείται μία στοίβα κλήσεων (call stack), στην οποία αποθηκεύονται διευθύνσεις μνήμης. Κάθε φορά που καλείται μία υπο-ρουτίνα (έστω GetAverage()), προστίθεται (push) στην στοίβα η διεύθυνση μνήμης της αμέσως επόμενης εντολής της ρουτίνας που την κάλεσε (έστω η CalculateStandardDeviation()). Εάν η CalculateStandardDeviation() αποτελείται από τις παρακάτω εντολές:

ΔΙΕΥΘΥΝΣΗ ΜΝΗΜΗΣ	ΕΝΤΟΛΗ
0	float CalculateStandardDeviation(int X[]){
1	int i,n=sizeof(X);
2	float SD=0;
3	for(i=0;i<n;i++){
4	Avg=GetAverage();
5	SD+=(X[i]-Avg)^2;
}	
6	SD/=n;
7	SD=sqrt(SD);
8	return SD;}

θέσεις μνήμης, όταν πραγματοποιείται η pop ελέγχεται εάν η διεύθυνση βρίσκεται εντός των ορίων που έχει θέσει το λειτουργικό σύστημα (με συγκεκριμένο μηχανισμό, ο οποίος δεν αφορά όμως το αντικείμενο της άσκησης).

Παράλληλα, αντί για την αποθήκευση της απόλυτης τιμής της διεύθυνσης, μπορεί να αποθηκεύεται η σχετική θέση της διεύθυνσης. Για χάρη απλότητας, θεωρήστε ότι αποθηκεύεται η διαφορά της τελευταίας εντολής της GetAverage() και της επόμενης εντολής της CalculateStandardDeviation(). Εάν θεωρήσουμε ότι η GetAverage() αποτελείται από τις παρακάτω εντολές:

ΔΙΕΥΘΥΝΣΗ ΜΝΗΜΗΣ	ΕΝΤΟΛΗ
9	float GetAverage(int X[]){
10	float Avg=0;
11	int n= sizeof(X);
12	for(i=0;i<n;i++){
13	Avg +=X[I;
}	
14	Avg /=n;
15	return Avg;}

Θα προστεθεί στην στοίβα η τιμή -10, ώστε (θέση μνήμης όταν τελειώσει η εκτέλεση)+(διαφορά)=(νέα θέση μνήμης), δηλαδή 15-10=5.

Γράψτε ένα πρόγραμμα, χρησιμοποιώντας την ΑΔΤ στοίβα, το οποίο προσομοιώνει τη λειτουργία αυτή ως εξής:

• Το πρόγραμμα δέχεται τη μέγιστη διεύθυνση μνήμης (έστω M = 100)

• Στη συνέχεια δέχεται σχετικές διευθύνσεις μνήμης, τις οποίες αποθηκεύει σε μία στοίβα και σταματά όταν δοθεί η τιμή 0 (καμία μετακίνηση).
• Ζητείται η τρέχουσα διεύθυνση μνήμης.
• Εκτελεί τις εντολές μία-μία, με τη σειρά που εξέρχονται από τη στοίβα, υπολογίζοντας την νέα διεύθυνση μνήμης και εκτυπώνοντας το μήνυμα (“Executing instruction: διεύθυνση μνήμης ”).
• Σε περίπτωση που η σχετική διεύθυνση οδηγεί σε μη επιτρεπτή θέση μνήμης (<0, ή >M) εκτυπώνεται το μήνυμα (“Access Violation Exception at address: διεύθυνση μνήμης ”) και σταματά η εκτέλεση.

```
Please enter maximum memory address: 100
Please enter the next relative memory address: -10
Please enter the next relative memory address: -20
Please enter the next relative memory address: 30
Please enter the next relative memory address: 0
Please enter the current memory address: 50
Executing instruction: 80
Executing instruction: 60
Executing instruction: 50
```

```
Please enter maximum memory address: 100
Please enter the next relative memory address: 120
Please enter the next relative memory address: -10
Please enter the next relative memory address: 30
Please enter the next relative memory address: 0
Please enter the current memory address: 50
Executing instruction: 80
Executing instruction: 70
Access Violation Exception at address: 190
```

```
Please enter maximum memory address: 100
Please enter the next relative memory address: -100
Please enter the next relative memory address: -20
Please enter the next relative memory address: 30
Please enter the next relative memory address: 0
Please enter the current memory address: 50
Executing instruction: 80
Executing instruction: 60
Access Violation Exception at address: -40
```