Unix Lab Assignment

GNU Project Debugger(GDB)

Name:G Saiteja
Rollno:422149
Sec: “A”

Code: (Finding circularity of a linked list)

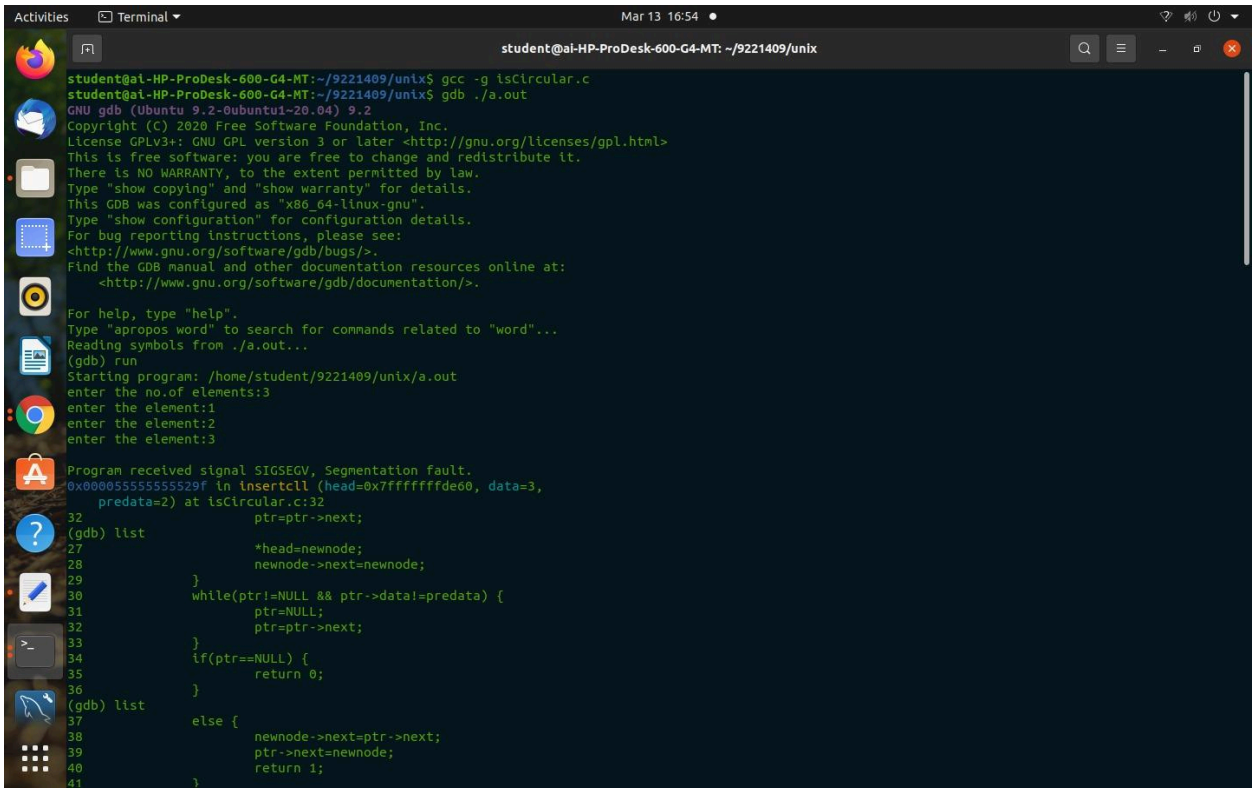
```
#include <stdio.h>
#include <stdlib.h>

struct node {
        int data;
        struct node*next;
        int visit;
};

int init(struct node**head) {
        *head=NULL;
        return 1;
}

int destroy(struct node**head) {
        *head=NULL;
        return 1;
}
int insertcll(struct node**head,int data,int predata) {
        struct node*newnode=(struct node*)malloc(sizeof(struct node));
        if(newnode==NULL) {
                return 0;
        }
```

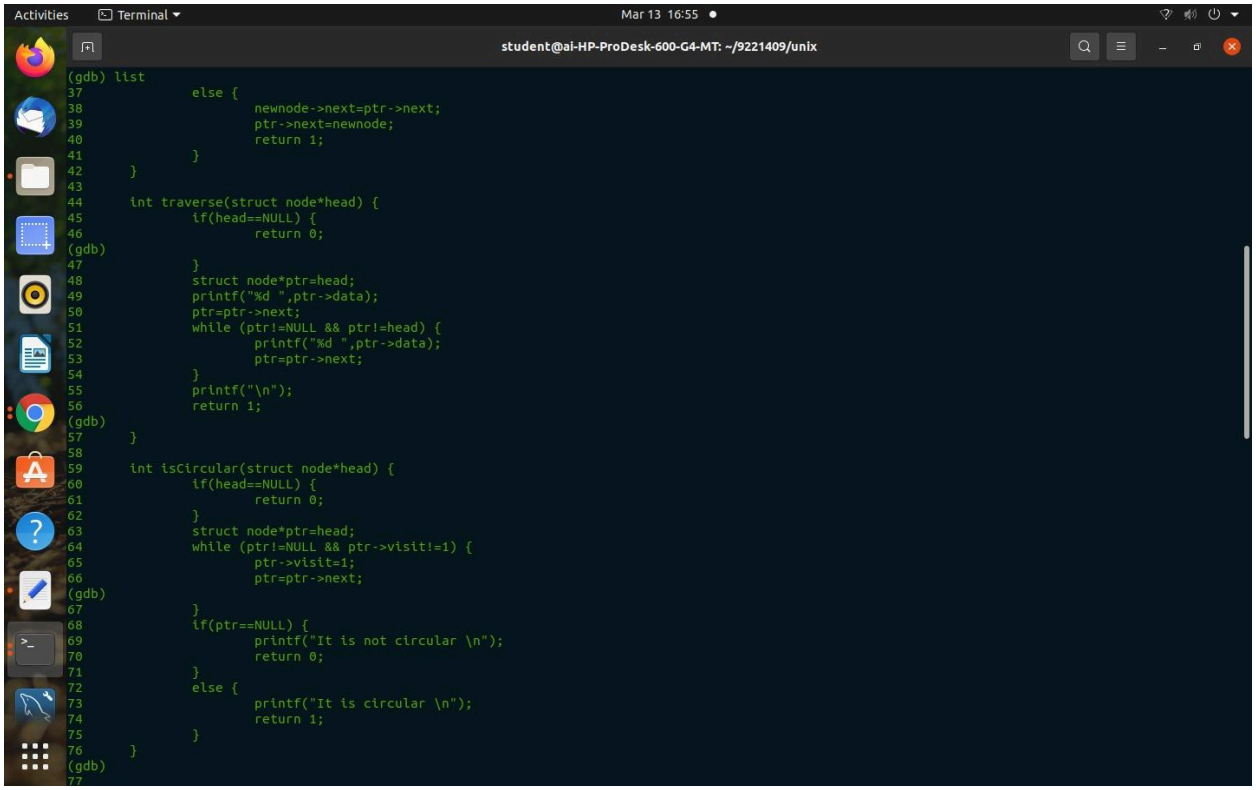
```
        newnode->data=data;
        struct node*ptr=*head;
        if(*head==NULL) {
                *head=newnode;
                newnode->next=newnode;
        }
        while(ptr!=NULL && ptr->data!=predata) {
                ptr=NULL;
                ptr=ptr->next;
        }
        if(ptr==NULL) {
                return 0;
        }
        else {
                newnode->next=ptr->next;
                ptr->next=newnode;
                return 1;
        }
}

int traverse(struct node*head) {
        if(head==NULL) {
                return 0;
        }
        struct node*ptr=head;
        printf("%d ",ptr->data);
        ptr=ptr->next;
        while (ptr!=NULL && ptr!=head) {
                printf("%d ",ptr->data);
                ptr=ptr->next;
        }
        printf("\n");
        return 1;
}
```

```
int isCircular(struct node*head) {
	if(head==NULL) {
		return 0;
	}
	struct node*ptr=head;
	while (ptr!=NULL && ptr->visit!=1) {
		ptr->visit=1;
		ptr=ptr->next;
	}
	if(ptr==NULL) {
		printf("It is not circular \n");
		return 0;
	}
	else {
		printf("It is circular \n");
		return 1;
	}
}

int main() {
	struct node* head =(struct node*)malloc(sizeof(struct node));
	init(&head);
	int n;
	int data1;
	int predata;
	printf("enter the no.of elements:");
	scanf("%d",&n);
	// inserting as circular linear linked lists
	for (int i=0;i<n;i++) {

		printf("enter the element:");
		scanf("%d",&data1);
	insertcll(&head,data1,predata);
		predata=data1;
	}
```

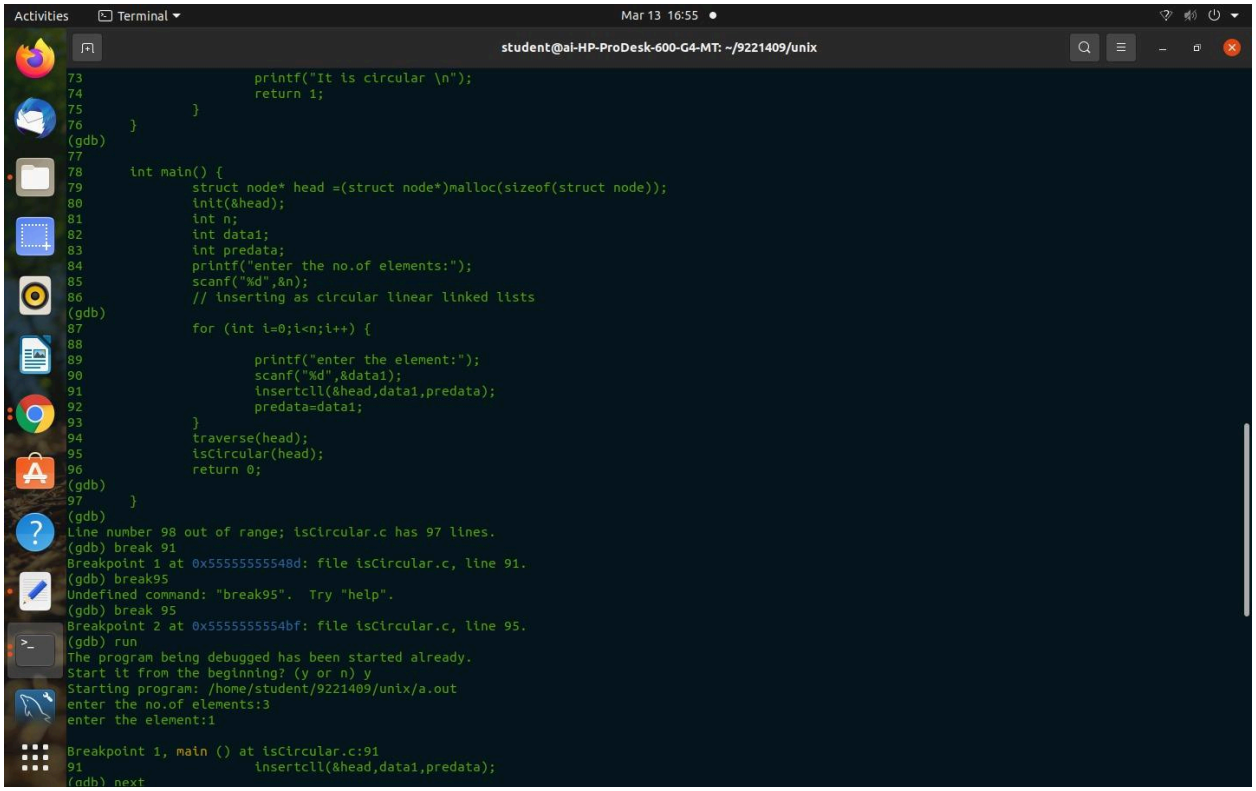
```
        traverse(head);
        isCircular(head);
        return 0;
}
```

Debugging:

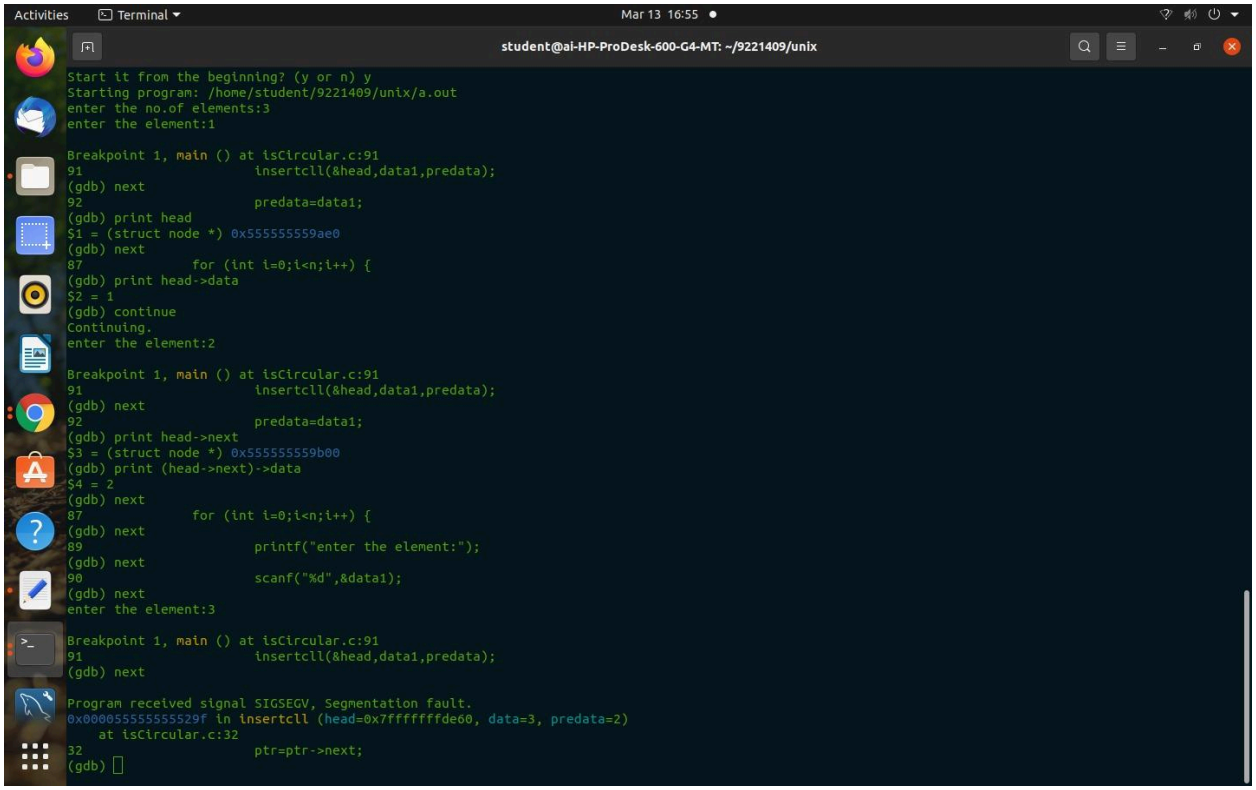





```
(gdb) list
37              else {
38                      newnode->next=ptr->next;
39                      ptr->next=newnode;
40                      return 1;
41              }
42      }
43
44      int traverse(struct node*head) {
45              if(head==NULL) {
46                      return 0;
(gdb)
47              }
48              struct node*ptr=head;
49              printf("%d ",ptr->data);
50              ptr=ptr->next;
51              while (ptr!=NULL && ptr!=head) {
52                      printf("%d ",ptr->data);
53                      ptr=ptr->next;
54              }
55              printf("\n");
56              return 1;
(gdb)
57      }
58
59      int isCircular(struct node*head) {
60              if(head==NULL) {
61                      return 0;
62              }
63              struct node*ptr=head;
64              while (ptr!=NULL && ptr->visit!=1) {
65                      ptr->visit=1;
66                      ptr=ptr->next;
(gdb)
67              }
68              if(ptr==NULL) {
69                      printf("It is not circular \n");
70                      return 0;
71              }
72              else {
73                      printf("It is circular \n");
74                      return 1;
75              }
76      }
(gdb)
77
```




```
73                      printf("It is circular \n");
74                      return 1;
75              }
76      }
(gdb)
77
78      int main() {
79              struct node* head =(struct node*)malloc(sizeof(struct node));
80              init(&head);
81              int n;
82              int data1;
83              int predata;
84              printf("enter the no.of elements:");
85              scanf("%d",&n);
86              // inserting as circular linear linked lists
(gdb)
87              for (int i=0;i<n;i++) {
88
89                      printf("enter the element:");
90                      scanf("%d",&data1);
91                      insertcll(&head,data1,predata);
92                      predata=data1;
93              }
94              traverse(head);
95              isCircular(head);
96              return 0;
(gdb)
97      }
(gdb)
Line number 98 out of range; isCircular.c has 97 lines.
(gdb) break 91
Breakpoint 1 at 0x55555555548d: file isCircular.c, line 91.
(gdb) break95
Undefined command: "break95".  Try "help".
(gdb) break 95
Breakpoint 2 at 0x5555555554bf: file isCircular.c, line 95.
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/student/9221409/unix/a.out
enter the no.of elements:3
enter the element:1

Breakpoint 1, main () at isCircular.c:91
91                      insertcll(&head,data1,predata);
(gdb) next
```



Segmentation fault in insertcll line number : 32

ptr=ptr->next;