Data structure practical 1:

Question 1:

Code :

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

typedef struct Mall {

        char name[20];
        int nShop;
        float rev;
        struct Mall *next;

}Mall;

Mall *head = NULL;

void addNode() {

        Mall *newNode = (Mall*)malloc(sizeof(Mall));
        printf("Enter mall name:");
        fgets(newNode->name,20,stdin);
        printf("Enter no of shops in mall:");
        scanf("%d",&newNode->nShop);
        printf("Enter revenue of mall:");
        scanf("%f",&newNode->rev);

        newNode->next = NULL;

        getchar();

        if(head == NULL)
                head = newNode;
        else {

                Mall *tmp = head;
                while(tmp->next != NULL)
                        tmp = tmp->next;

                tmp->next = newNode;
        }
}

void printList() {

        Mall *tmp = head;

        while(tmp != NULL) {

                printf("|%s_%d_%f|->",tmp->name,tmp->nShop,tmp->rev);
                tmp=tmp->next;
```

```
        }

        printf("NULL\n");
    }

    void main() {

        addNode();
        printList();
        addNode();
        printList();
    }
```

Output:

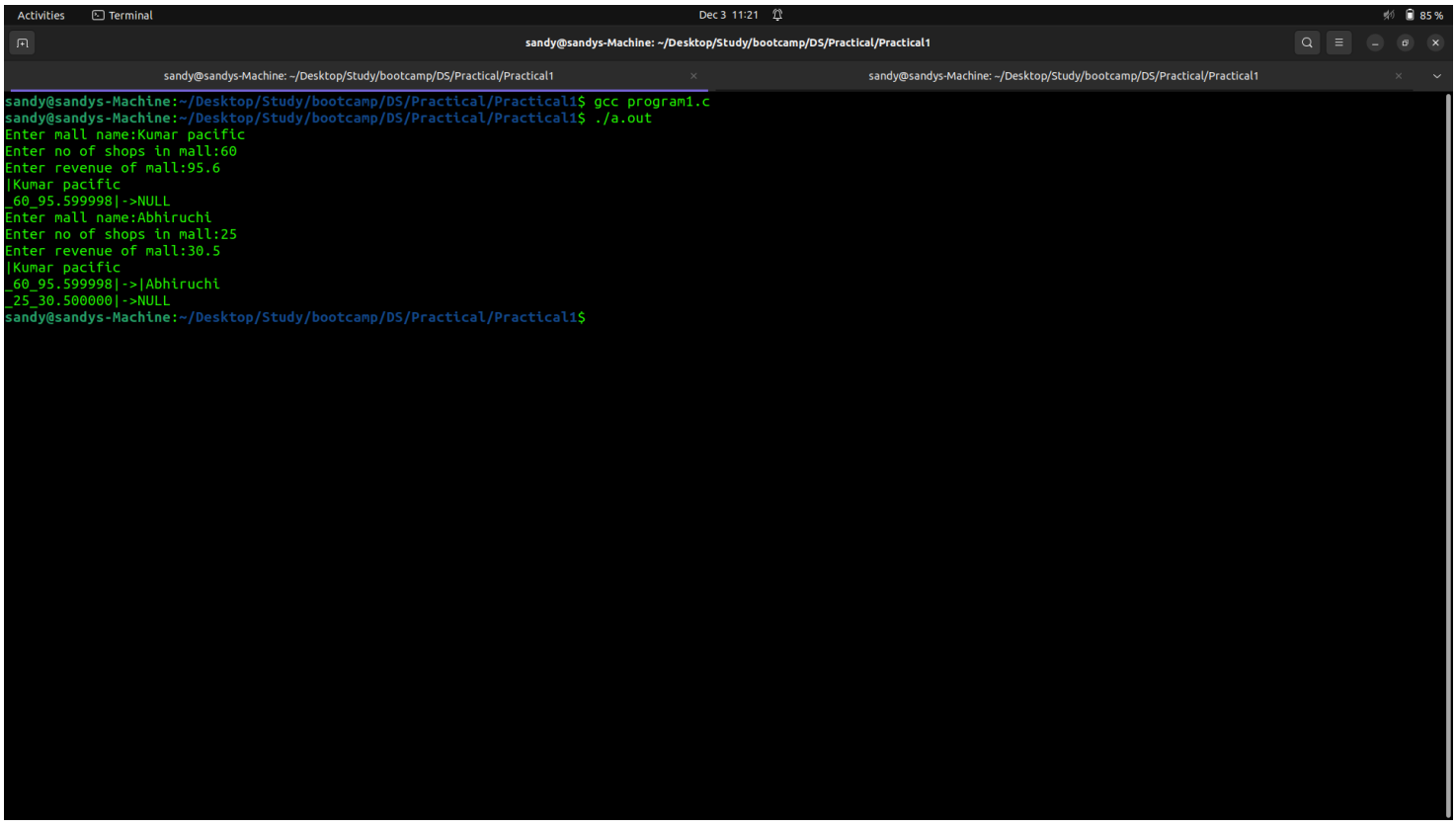
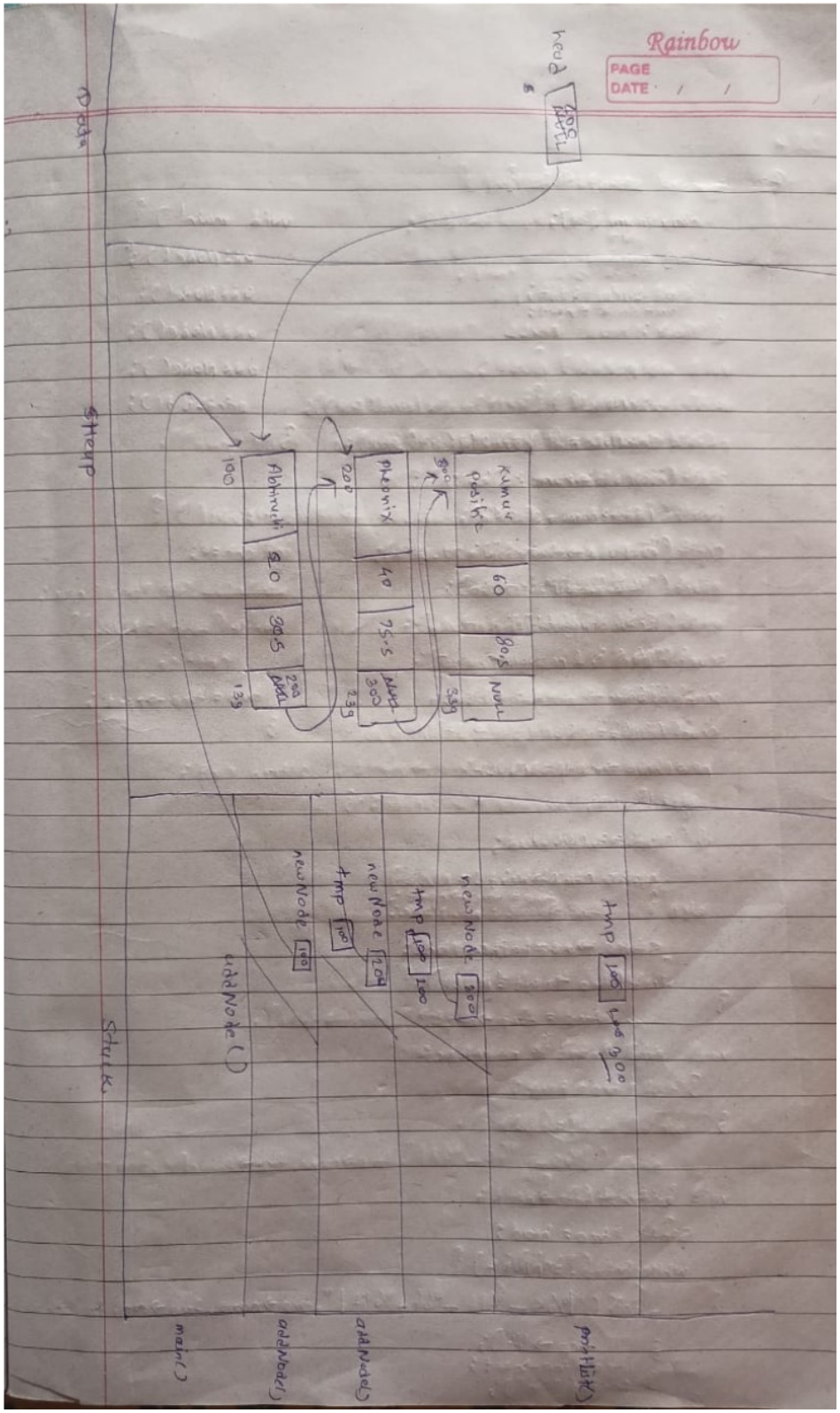


Diagram :


Rainbow
PAGE
DATE
head
200
Data
Steap
100
Abhinav
80
995
200
139
202
Phoenix
40
755
NULL
300
239
300
Kumar
Sahil
60
805
NULL
339
addNode()
newNode
100
tmp
100
addNode()
newNode
200
tmp
200
addNode()
newNode
300
tmp
300
printlist()
tmp
100
Link 200
main()
Stack

Question 2:

Code :

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

typedef struct State {

	char name[20];
	int pop;
	float budget,lit;
	struct State *next;

}State;

State *head = NULL;

void addNode() {

	State *newNode = (State*)malloc(sizeof(State));
	printf("Enter state name:");
	fgets(newNode->name,20,stdin);
	printf("Enter population:");
	scanf("%d",&newNode->pop);
	printf("Enter budget of state:");
	scanf("%f",&newNode->budget);
	printf("Enter lit rate of state:");
	scanf("%f",&newNode->lit);

	newNode->next = NULL;

	getchar();

	if(head == NULL)
		head = newNode;
	else {

		State *tmp = head;
		while(tmp->next != NULL)
			tmp = tmp->next;

		tmp->next = newNode;
	}
}

void printList() {

	State *tmp = head;

	while(tmp != NULL) {

		printf("|%s_%d_%f_%f|->",tmp->name,tmp->pop,tmp->budget,tmp->lit);
		tmp=tmp->next;
	}
```

```
        printf("NULL\n");
}

void main() {

        addNode();
        addNode();
        addNode();
        printList();
}
```

Output:

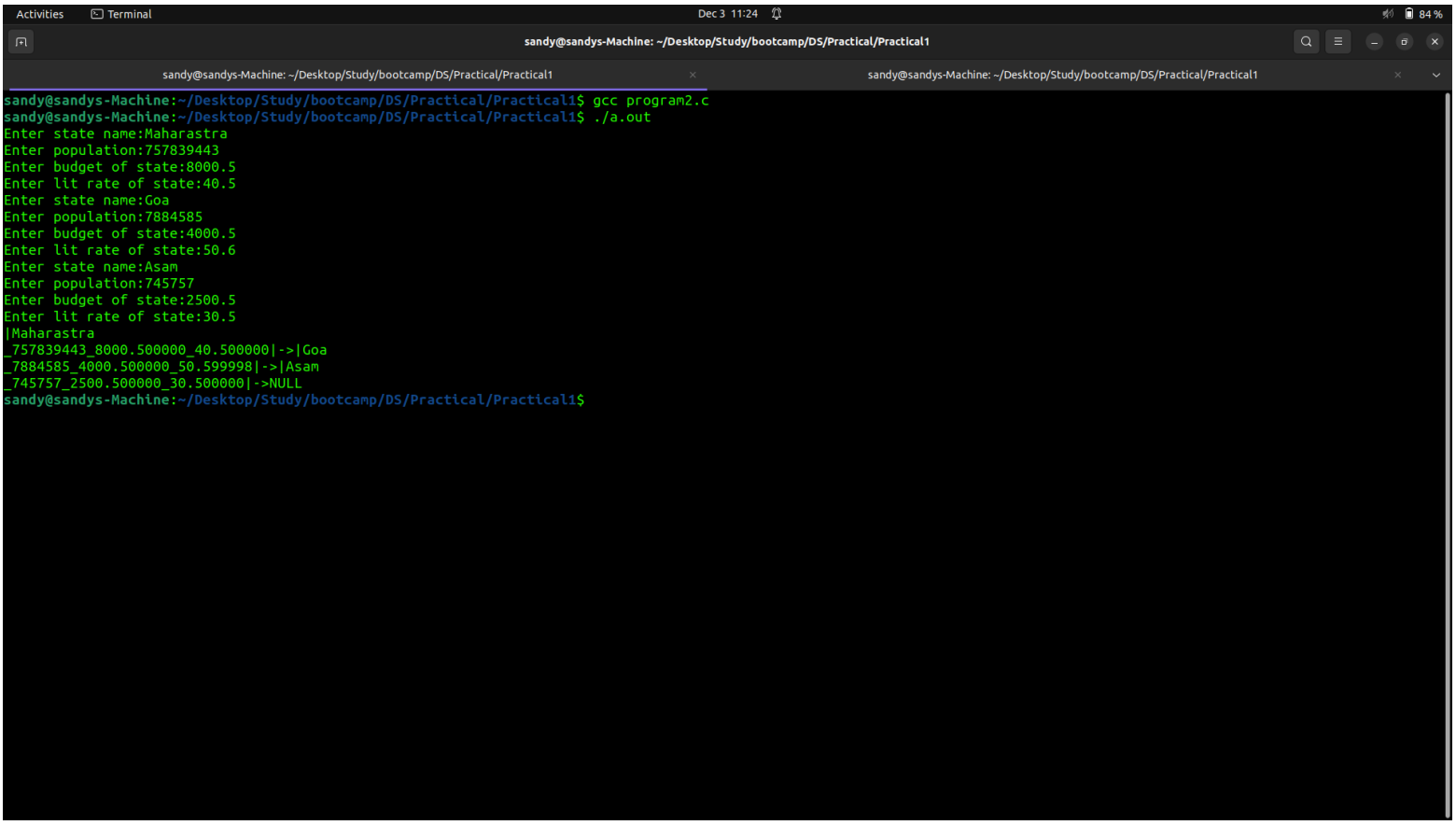
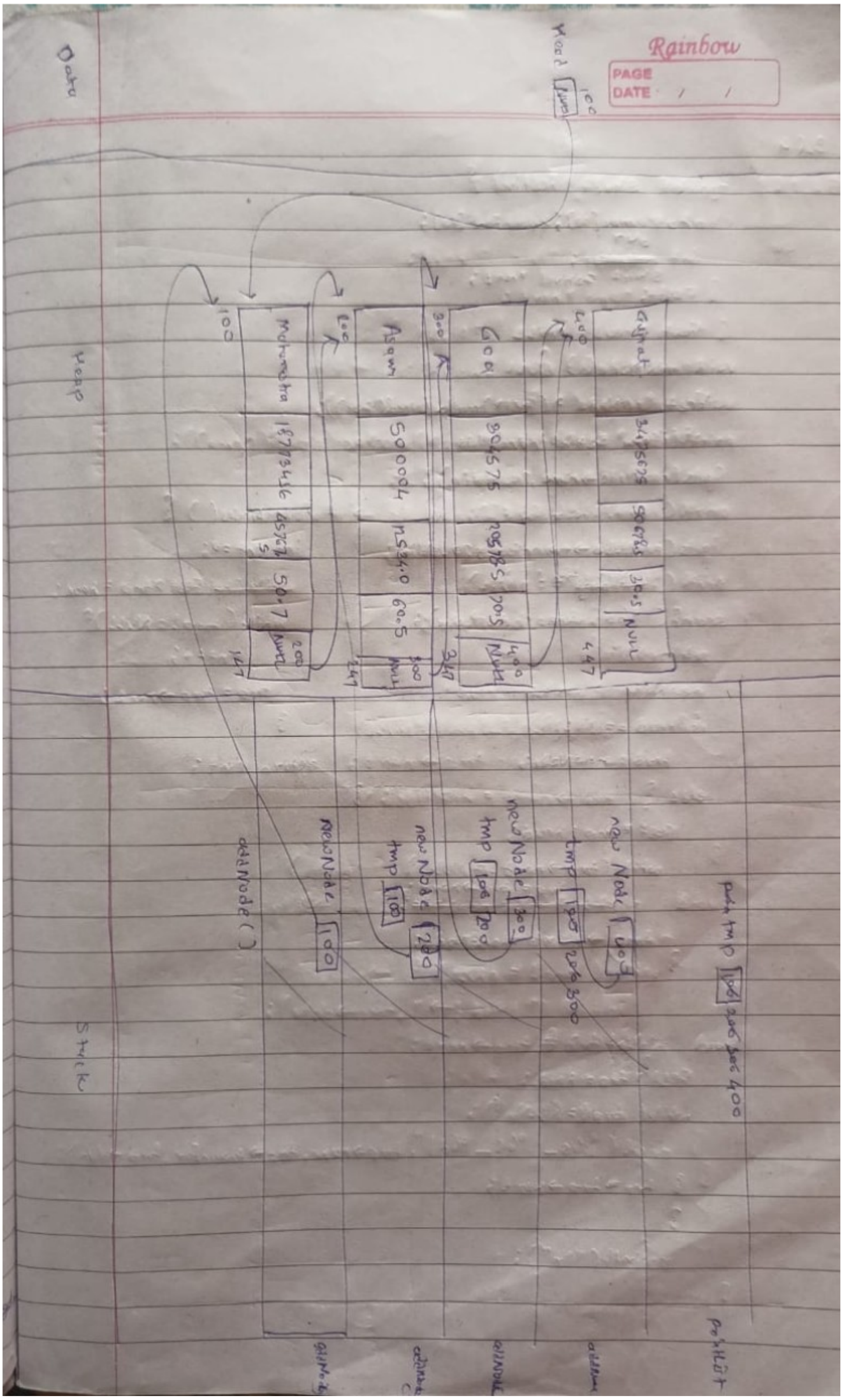


Diagram:


Rainbow
PAGE
DATE
Data
Heap
Stack
Gujrat
Goa
Assam
Maharashtra
NULL

Question 3 & 4:

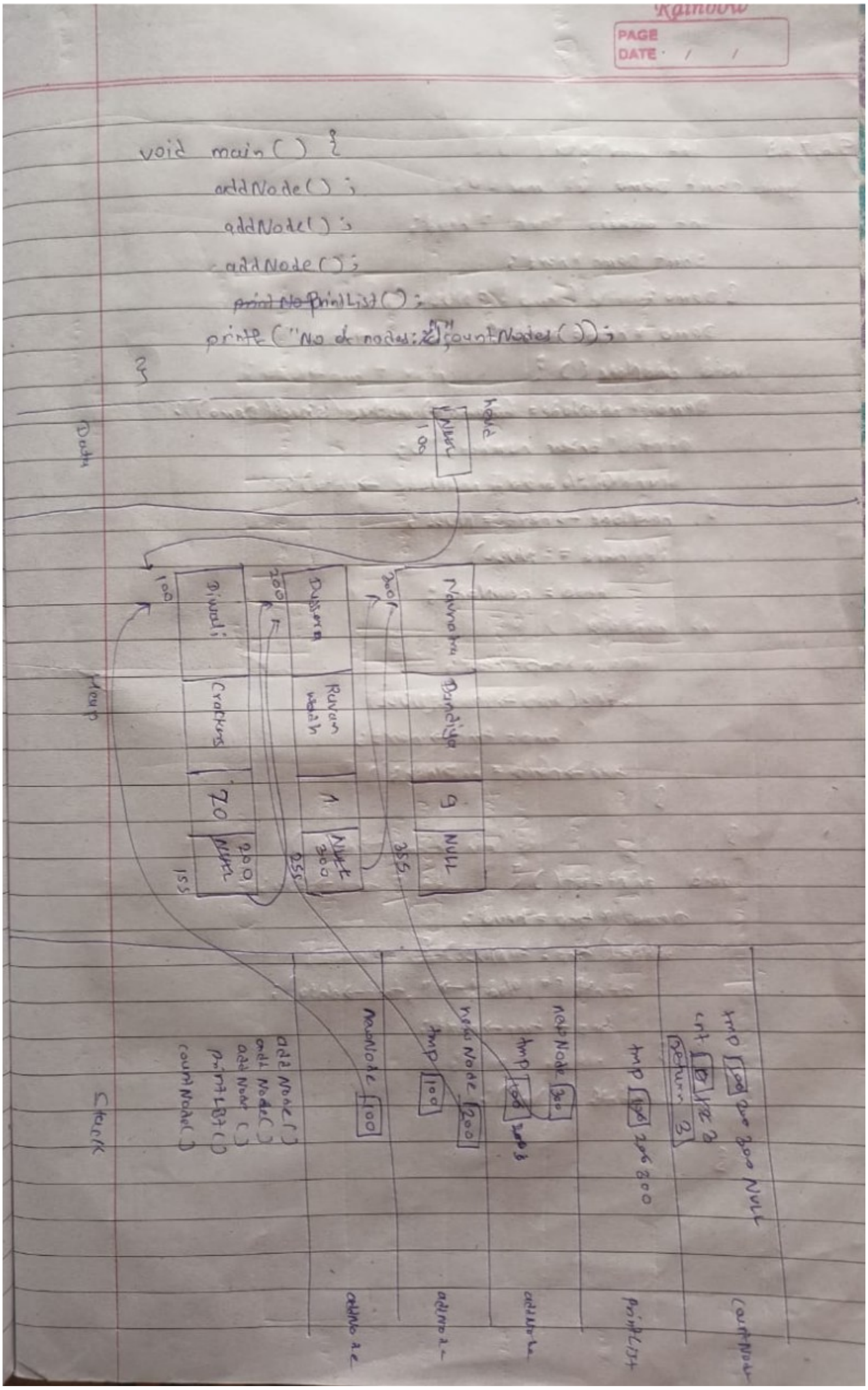
Code :

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

typedef struct Fest {

        char name[20],knownFor[20];
        int dur;
        struct Fest *next;

}Fest;

Fest *head = NULL;

void addNode() {

        Fest *newNode = (Fest*)malloc(sizeof(Fest));
        printf("Enter festival name:");
        fgets(newNode->name,20,stdin);
        printf("Enter speciality of festival:");
        fgets(newNode->knownFor,20,stdin);
        printf("Enter duration of festival(days):");
        scanf("%d",&newNode->dur);

        newNode->next = NULL;

        getchar();

        if(head == NULL)
                head = newNode;
        else {

                Fest *tmp = head;
                while(tmp->next != NULL)
                        tmp = tmp->next;

                tmp->next = newNode;
        }
}

void printList() {

        Fest *tmp = head;

        while(tmp != NULL) {

                printf("|%s_%s_%d|->",tmp->name,tmp->knownFor,tmp->dur);
                tmp=tmp->next;
        }

        printf("NULL\n");
}
```

```
int countNodes() {

        int cnt=0;
        Fest *tmp = head;

        while(tmp!=NULL) {

                cnt++;
                tmp = tmp->next;
        }

        return cnt;
}

void main() {

        addNode();
        addNode();
        printList();

        printf("\nNo of nodes in list are:%d\n",countNodes());
}
```

Output:

Diagram :



Question 5 ,6, 7, 8, 9 :

Code :

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

typedef struct Demo {

	int data;
	struct Demo *next;

}Demo;

Demo *head = NULL;

void addNode() {

	Demo *newNode = (Demo*)malloc(sizeof(Demo));

	printf("Enter data:");
	scanf("%d",&newNode->data);

	newNode->next = NULL;

	if(head == NULL)
		head = newNode;
	else {

		Demo *tmp = head;
		while(tmp->next != NULL)
			tmp = tmp->next;

		tmp->next = newNode;
	}
}

void printList() {

	Demo *tmp = head;

	while(tmp != NULL) {

		printf("|%d|->",tmp->data);
		tmp=tmp->next;
	}

	printf("NULL\n");
}

int nodeSum() {

	int sum=0;
	Demo *tmp = head;
```

```
	while(tmp != NULL) {

		sum+=tmp->data;
		tmp = tmp->next;
	}

	return sum;
}

int firstNLastSum() {

	int sum=0;
	Demo *tmp = head;

	if(tmp == NULL) {
		return -1;
	} else if(tmp->next == NULL) {

		return tmp->data;
	}else {
		sum+=tmp->data;
	}

	while(tmp->next != NULL)
		tmp = tmp->next;

	sum+=tmp->data;

	return sum;
}

int maxData() {

	int max=0;

	Demo *tmp = head;

	if(tmp == NULL) {
		return -1;
	} else if(tmp->next == NULL) {
		return tmp->data;
	}

	while(tmp != NULL) {

		if(max < tmp->data)
			max = tmp->data;

		tmp = tmp->next;
	}

	return max;
}

int minData() {
```

```
	int min=999999;
	Demo *tmp = head;

	if(tmp == NULL) {
		return -1;
	} else if(tmp->next == NULL) {
		return tmp->data;
	}

	while(tmp != NULL) {

		if(min > tmp->data)
			min = tmp->data;

		tmp = tmp->next;
	}

	return min;
}

bool isPrime(int n) {

	int cnt=0;

	for(int i=2;i<n;i++) {

		if(n%i==0)
			cnt++;
	}

	if(cnt <= 1)
		return true;

	return false;
}

bool isPrimePresent() {

	Demo *tmp = head;

	while(tmp != NULL) {

		if(isPrime(tmp->data))
			return true;

		tmp = tmp->next;
	}

	return false;
}



void main() {

	int n;
	printf("How much nodes you want to create:");
```

```
        scanf("%d",&n);

        for(int i=0;i<n;i++)
                addNode();

        printList();

        printf("Sum of data of all nodes:%d\n",nodeSum());
        printf("Sum of data of first and last nodes:%d\n",firstNLastSum());
        printf("Maximum data in list is %d\n",maxData());
        printf("Minimum data in list is %d\n",minData());

        if(isPrimePresent())
                printf("Prime no is present in list\n");
        else
                printf("Prime no is not present in list\n");
}
```

Output:

Question 10:

Code :

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

typedef struct FootballWC {

	int noOfTeams;
	char hostCountry[20],bestStriker[20],bestDefender[20];
	float budget;
	struct FootballWC *next;

}FWC;

FWC *head = NULL;

void addNode() {

	FWC *newNode = (FWC*)malloc(sizeof(FWC));
	printf("Enter no of teams in Fifa world cup:");
	scanf("%d",&newNode->noOfTeams);

	getchar();

	printf("Enter host country name:");
	fgets(newNode->hostCountry,20,stdin);
	printf("Enter best striker name:");
	fgets(newNode->bestStriker,20,stdin);
	printf("Enter best defender name:");
	fgets(newNode->bestDefender,20,stdin);
	printf("Enter budget of football world cup:");
	scanf("%f",&newNode->budget);

	newNode->next = NULL;

	if(head == NULL)
		head = newNode;
	else {

		FWC *tmp = head;
		while(tmp->next != NULL)
			tmp = tmp->next;

		tmp->next = newNode;
	}
}

void printList() {

	FWC *tmp = head;

	while(tmp != NULL) {
```

```
            printf("|%d_%s_%s_%s_%f|->",tmp->noOfTeams,tmp->hostCountry,
            tmp->bestStriker,tmp->bestDefender,tmp->budget);
            tmp=tmp->next;
        }

        printf("NULL\n");
}

void main() {

        addNode();
        addNode();
        printList();
}
```

Output:

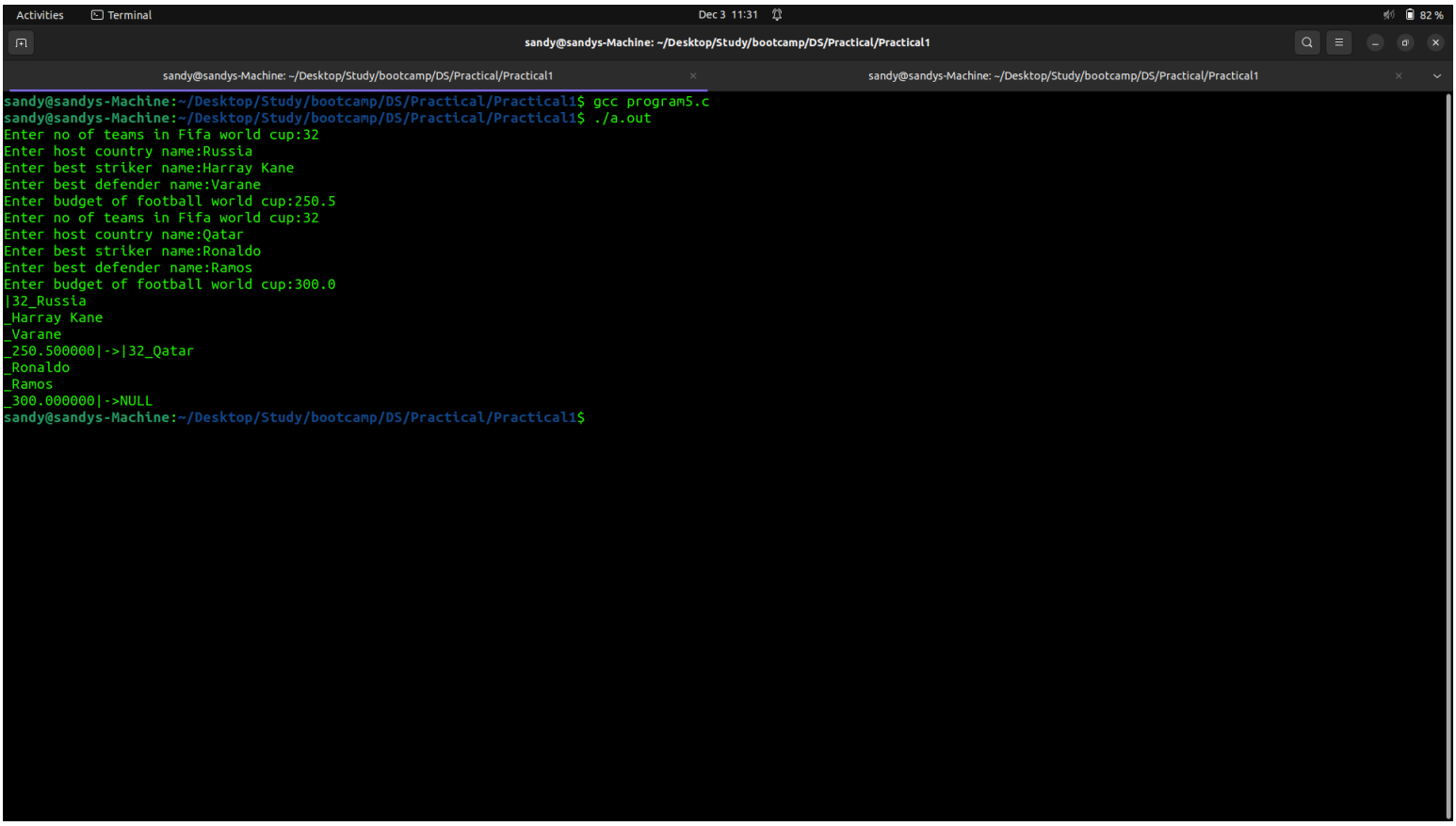
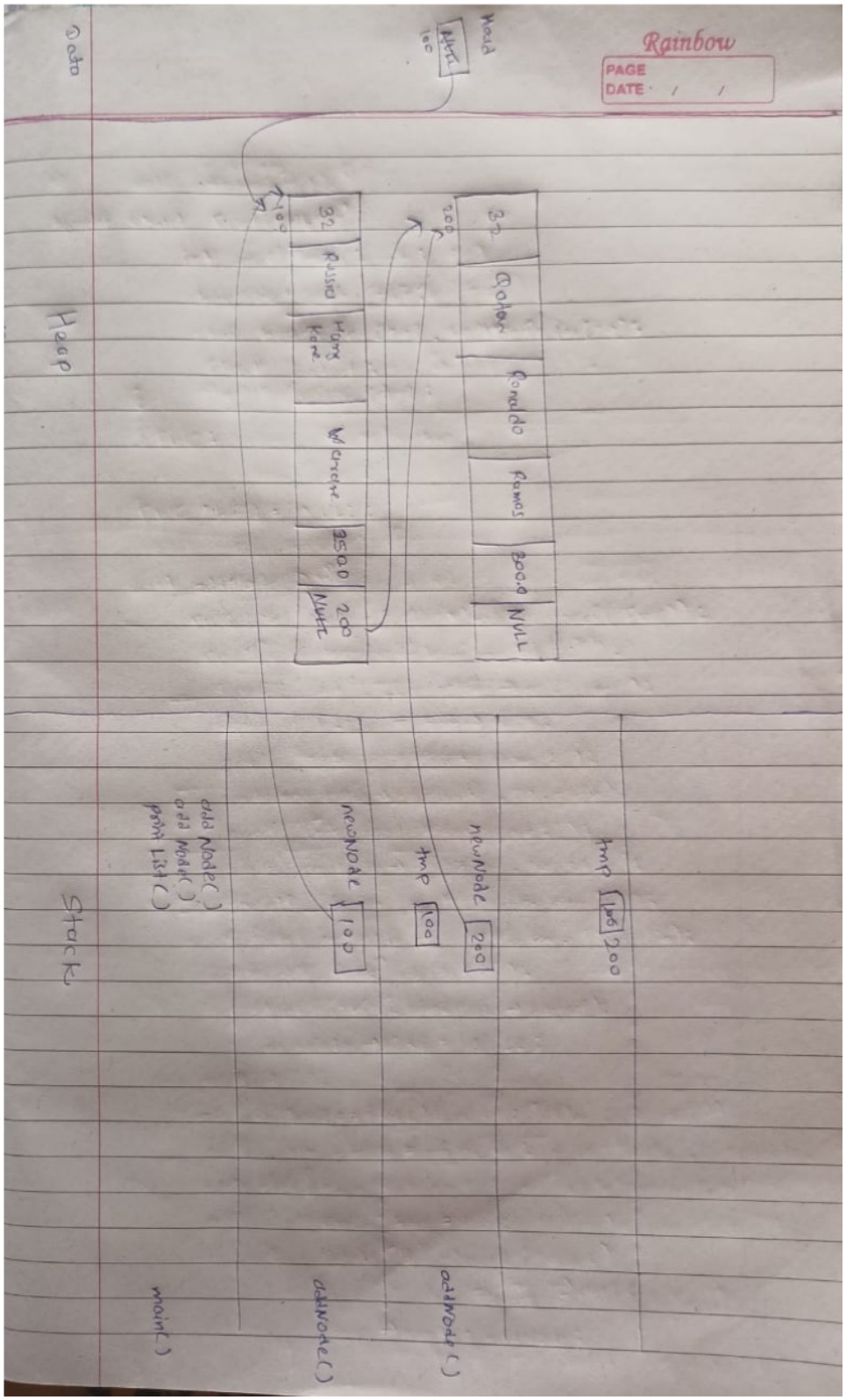


Diagram:


Rainbow
PAGE
DATE / /
Data
Head
Addr
100
Heap
32
Qatar
Ronaldo
Points
200.0
NULL
200
32
Russia
Hong Kong
Wrestler
2500
200
Addr
100
Stack
addNode()
printList()
main()
tmp
100
200
newNode
200
tmp
100
newNode
100
addNode()