Debugging Assignment

Q1.

#include int main() {

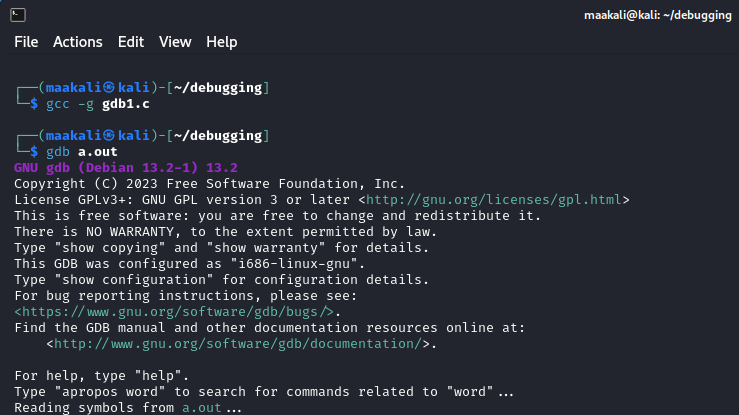
int a,b;

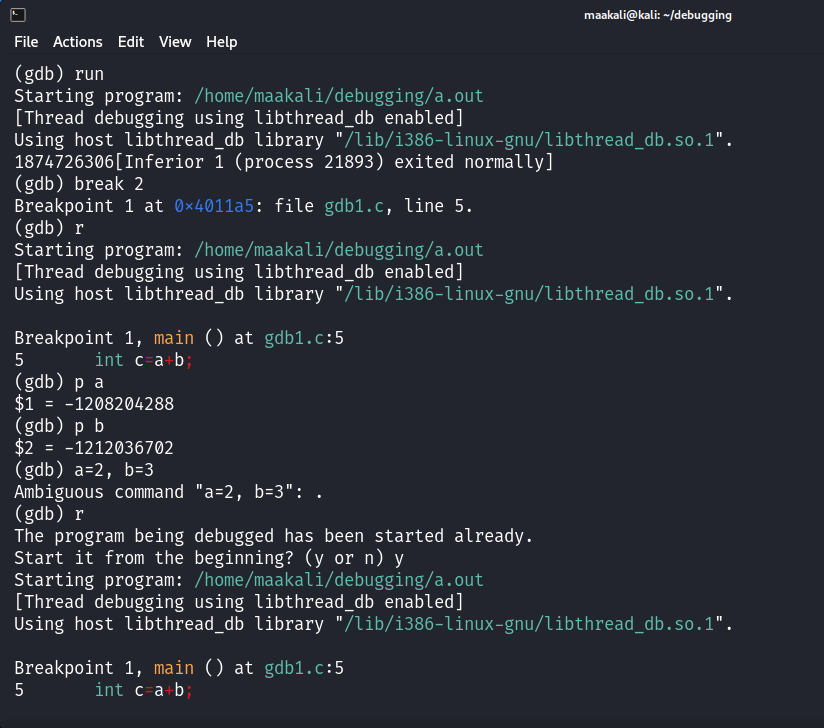
int c=a+b;

printf("%d",c); // Output: 5

return 0;

}





Q.2)

#include int main() {

int d=2;

printf("Enter the value of d:");

scanf("%d",d);

printf("The value of d is:%d",d);

return 0;

}

Correct code

#include <stdio.h>

int main()

{

int d=2;

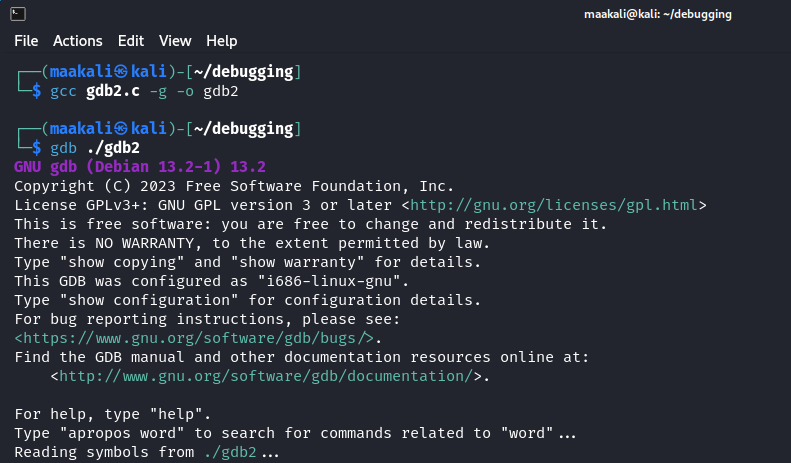
printf("Enter the value of d:");

scanf("%d",&d);

printf("The value of d is:%d",d);

return 0;

}





Q.3)

#include <stdio.h>

#include <stdlib.h>

int factorial(int n);

int main(void) {

int n = 5;

int f = factorial(n);

printf("The factorial of %d is %d.\n", n, f); n = 17;

f = factorial(n);

printf("The factorial of %d is %d.\n", n, f); return 0;

}

//A factorial is calculated by n! = n \* (n - 1) \* (n - 2) \* ... \* 1 //E.g. 5! = 5 \* 4 \* 3 \* 2 \* 1 = 120

int factorial(int n)

{ int f = 1; int i = 1;

while (i <= n) { f = f \* i; i++; }

return f;

}

Correct code Q.3

#include <stdio.h>

#include <stdlib.h>

long long int factorial(int n);

int main(void) {

int n = 5;

long long int f = factorial(n);

printf("The factorial of %d is %lld.\n", n, f);

n = 17;

f = factorial(n);

printf("The factorial of %d is %lld.\n", n, f);

return 0;

}

//A factorial is calculated by n! = n \* (n - 1) \* (n - 2) \* ... \* 1

//E.g. 5! = 5 \* 4 \* 3 \* 2 \* 1 = 120

long long int factorial(int n) {

long long int f = 1;

int i = 1;

while (i <= n) {

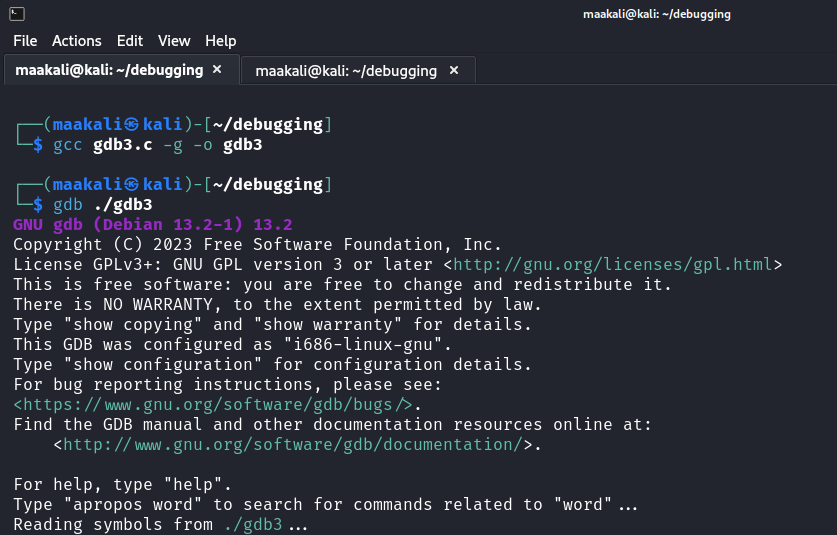
f = f \* i;

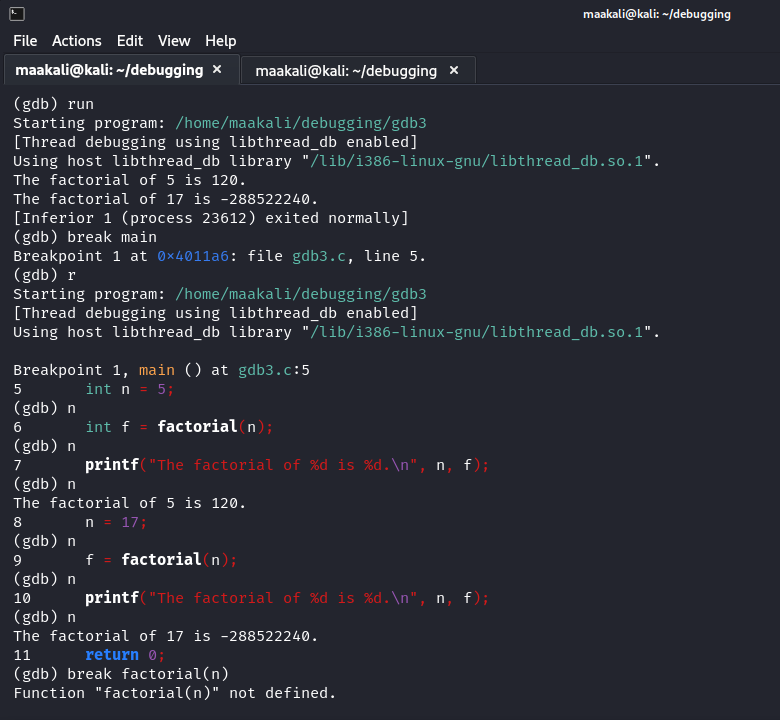
i++;

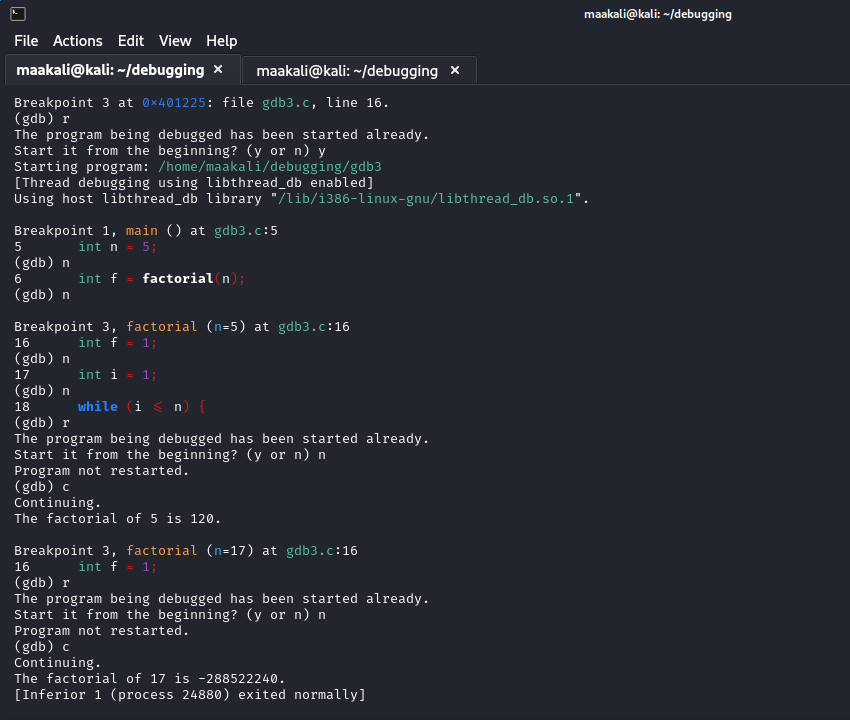
}

return f;

}

****

****

****

**Q.4)**

#include <stdio.h>

int main(void) {

int arr[2];

arr[3] = 10; // Accessing out of bound

return (0);

}

Correct code for Q.4):-

#include <stdio.h>

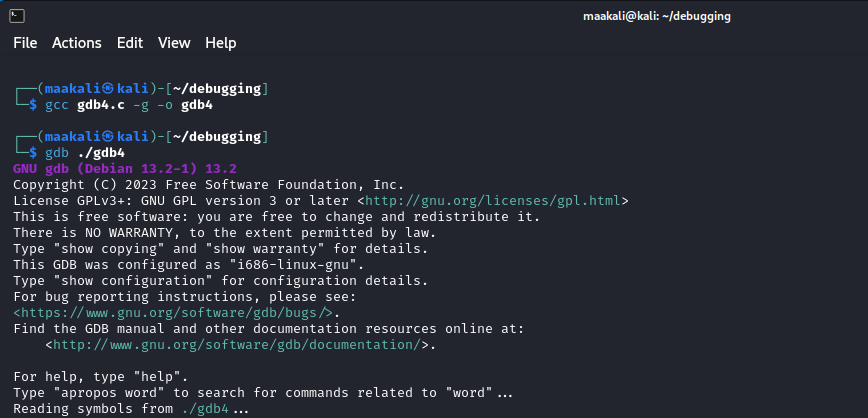
int main(void) {

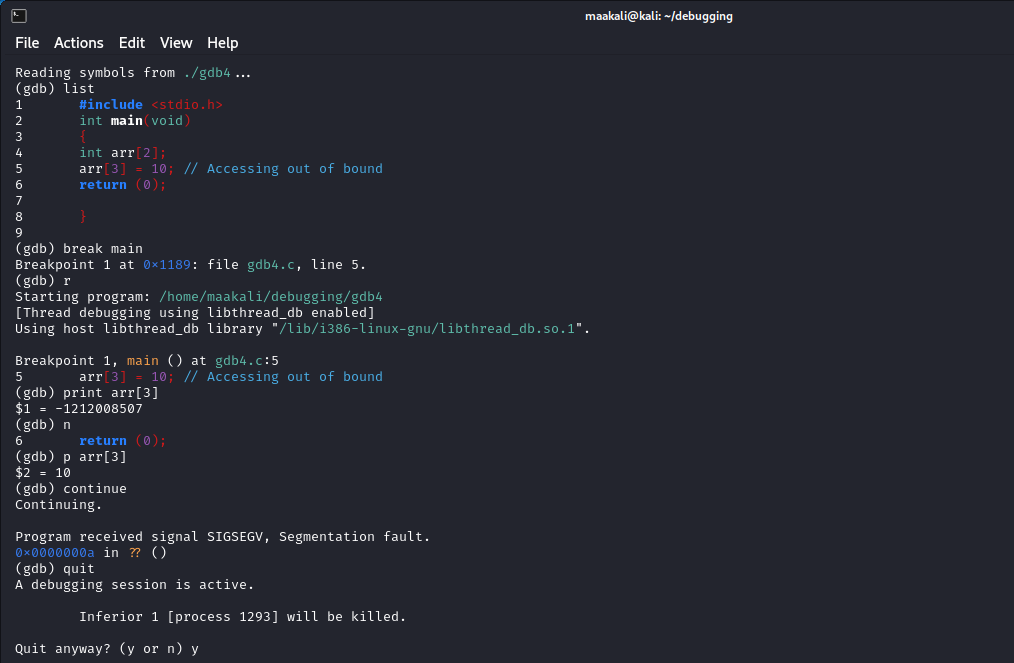
int arr[4];

arr[3] = 10;

return (0);

}



****

**Q.5)**

#include <stdio.h>

int main(){

int \*p;

printf("%d",\*p);

return 0;

}

Correct code Q.5):-

#include <stdio.h>

int main()

{

int \*p;

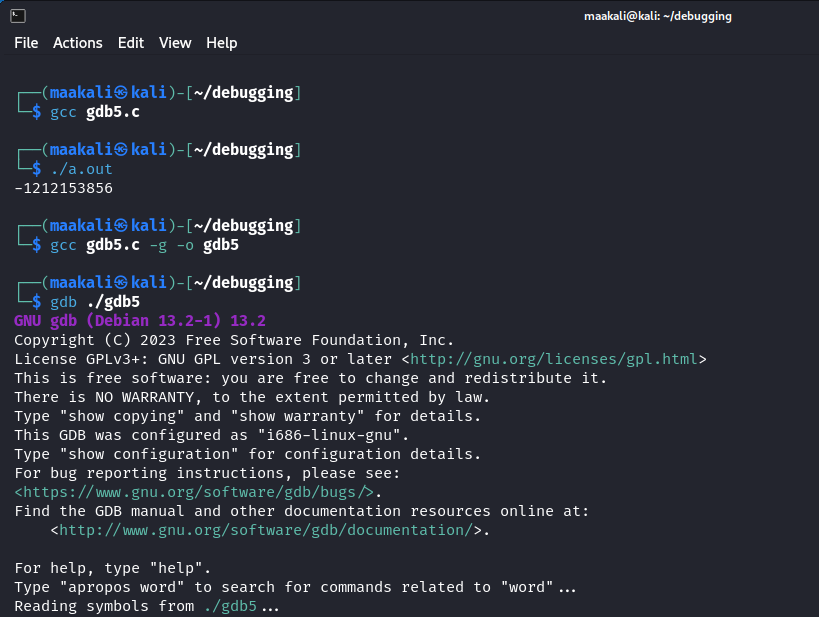
int var = 15;

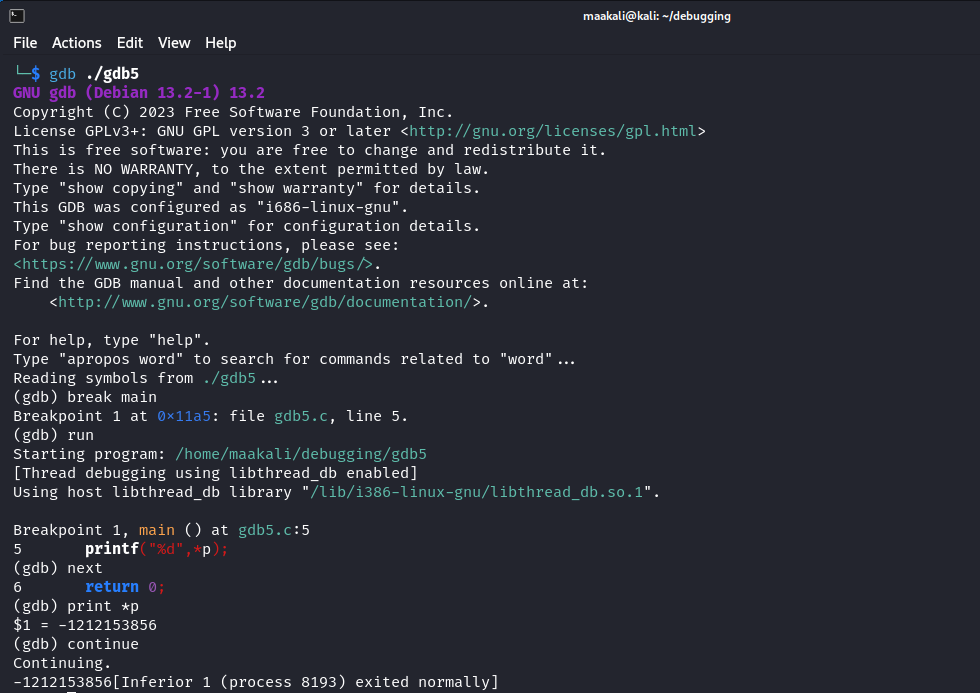
p = &var;

printf("%d",\*p);

return 0;

}

****

****