PRACTISE

srilatha@GESLMP22WP7T:~/Experiments/misc/practise$ cd arrays/

srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays$ ls

arrprog.c arrsum output primearr seats

srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays$ cat arrprog.c

```
#include<stdio.h>
#if 1 //12
	int main()
	{
		int num;
		int j;
		int i;
		int pos;
		int ele;
		int arr[100];
		int temp;
		printf("Enter number of elements in array:");
		scanf("%d",&num);
		printf("Enter elements of array:");
		for(i = 0; i < num; i++) {
			scanf("%d",&arr[i]);
		}
		printf("Enter the element to insert:");
		scanf("%d",&ele);
		printf("Enter the position to insert element:");
		scanf("%d",&pos);
		for(i = 0; i < num; i++) {
			if(pos == i) {
				for(j = i;j < num+1; j++) {
```

```
                temp = arr[i];
                arr[j]=ele;
                arr[j+1]=temp;
                }
            }
        }
        for(i = 0; i <= num + 1; i++) {
            printf("%d",arr[i]);
        }
        return 0;
        }
#endif
#if 0 //11
    int main()
    {
        int num;
        int i;
        int j;
        int temp;
        int arr[100];
        printf("Enter number of elements in array:");
        scanf("%d",&num);
        printf("Enter elements of array:");
        for(i = 0; i < num; i++) {
            scanf("%d",&arr[i]);
        }
        for(i = 0; i < num; i++) {
            for(j = i+1; j < num; j++) {
                if(arr[j] > arr[i]) {
```

```
                    temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }
        printf("Array elements after sorting is:");
        for( i = 0; i < num; i++) {
            printf("%d\n",arr[i]);
        }
        return 0;
        }
#endif
#if 0 //10
    int main()
    {
        int num;
        int i;
        int j;
        int temp;
        int arr[100];
        printf("Enter number of elements in array:");
        scanf("%d",&num);
        printf("Enter elements of array:");
        for(i = 0; i < num; i++) {
            scanf("%d",&arr[i]);
        }
        for(i = 0; i < num; i++) {
            for(j = i+1; j < num; j++) {
```

```
                if(arr[j] < arr[i]) {
                    temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }
        printf("Array elements after sorting is:");
        for( i = 0; i < num; i++) {
            printf("%d",arr[i]);
        }
        return 0;
        }
#endif

#if 0 //9
    int main()
    {
        int num;
        int arr[10];
        int i;
        int key;
        int count = 0;
        printf("Enter the number of array elements:");
        scanf("%d",&num);
        printf("Enter array elements:");
        for(i = 0; i < num; i++) {
            scanf("%d", &arr[i]);
        }
```

```
        printf("Enter the element to search in array: ");
        scanf("%d", &key);
        for(i = 0; i < num; i++) {
            if(arr[i] == key) {
                printf("number found at %d index \n",i);
                count++;
            }
        }
        if(count == 0) {
            printf("Element not found in array\n");
        }
        return 0;
        }
#endif

#if 0 //8
    int main()
    {
        int matrix1[50][50];
        int matrix2[50][50];
        int mat3[10][10];
        int rows;
        int cols;
        int i;
        int j;
        int k;
        printf("Enter the number of rows in matrix :");
        scanf("%d", &rows);
        printf("Enter the number of cols in matrix :");
```

```
        scanf("%d", &cols);
        printf("Enter the elements of matrix1 :");
        for(i = 0; i < rows; i++) {
            for(j = 0; j < cols; j++) {
                scanf("%d",&matrix1[i][j]);
                }
        }
        printf("Enter the elements of matrix2 :");
        for(i = 0; i < rows; i++) {
            for(j = 0; j < cols; j++) {
                scanf("%d",&matrix2[i][j]);
                }
        }
        printf("multiplication of matrices is:");

        for(i = 0; i < rows; i++) {
            for(j = 0; j < cols; j++) {
                mat3[i][j] = 0;
                for(k = 0; k < cols; k++) {
                    mat3[i][j] = mat3[i][j] + matrix1[i][k] * matrix2[k][j];
                }
            printf("%d  ",mat3[i][j]);
            }
        printf("\n");
        }
        return 0;
        }
#endif
```

```
#if 0 //7
	int main()
	{
		int matrix1[50][50];
		int matrix2[50][50];
		int rows;
		int cols;
		int i;
		int j;
		printf("Enter the number of rows in matrix :");
		scanf("%d", &rows);
		printf("Enter the number of cols in matrix :");
		scanf("%d", &cols);
		printf("Enter the elements of matrix1 :");
		for(i = 0; i < rows; i++) {
			for(j = 0; j < cols; j++) {
				scanf("%d",&matrix1[i][j]);
				}
		}
		printf("Enter the elements of matrix2 :");
		for(i = 0; i < rows; i++) {
			for(j = 0; j < cols; j++) {
				scanf("%d",&matrix2[i][j]);
				}
		}
		printf("Addition of two matrices result is:\n");
		for(i = 0; i < rows; i++) {
			for(j = 0; j < cols; j++) {
```

```
                printf("%d ", matrix1[i][j] + matrix2[i][j]);
            }
        printf("\n");
        }
        return 0;
        }
#endif

#if 0//6
    int main()
    {
        int num;
        int arr[50];
        int arrneg[50];
        int i;
        int temp = 0;
        printf("Enter the number of array elements:");
        scanf("%d",&num);
        for(i = 0; i < num; i++) {
            scanf("%d",&arr[i]);
        }
        for(i = 0; i < num; i++) {
            if(arr[i] > temp) {
                temp = arr[i];
            }
        }
        printf("Largest number in array is : %d\n",temp);

        for(i = 0; i < num; i++) {
```

```
            if(arr[i] < temp) {
                temp = arr[i];
            }
        }
        printf("smallest number in array is : %d\n",temp);

        return 0;
        }
#endif

#if 0//5
    int main()
    {
        int num;
        int arr[50];
        int arrneg[50];
        int i;
        int j = 0;
        printf("Enter the number of array elements:");
        scanf("%d",&num);
        for(i = 0; i < num; i++) {
            scanf("%d",&arr[i]);
        }
        for(i = 0; i < num; i++) {
            if(arr[i] < 0) {
                arrneg[j] = arr[i];
                j++;
            }
        }
```

```
        printf("\nNegative numbers in array are %d:",j);
        if(j > 1) {
        printf("\nThey are : ");
            for(i = 0; i < j; i++) {
                printf("%d\n\t   ",arrneg[i]);
            }
        }
        return 0;
        }
#endif



#if 0//4
    int main()
    {
        int num;
        int arr[10];
        int sum=0;
        int i;
        printf("Enter the number of array elements:");
        scanf("%d ",&num);
        printf("Enter array elements:");
        for(i = 0; i <= num; i++) {
            scanf("%d", &arr[i]);
        }
        for(i = 0; i <= num; i++) {
            sum = sum + arr[i];
        }
        printf("sum of array elements is :%d", sum);
```

```
        return 0;
    }
#endif



#if 0//3
    int main()
    {
        int num;
        int arr[10];
        int i;
        printf("Enter the number of array elements:");
        scanf("%d",&num);
        printf("Enter array elements:");
        for(i = 0; i <= num; i++) {
            scanf("%d", &arr[i]);
        }
        printf("array elements in reverse order are:");
        for(i = num; i >= 0; i--) {
            printf("%d ", arr[i]);
        }
    return 0;
    }
#endif

#if 0//2
    int main()
    {
        int num;
```

```
        int arr[10];
        int count = 0;
        int i;
        int j;
        printf("Enter the number of array elements:");
        scanf("%d",&num);
        printf("Enter array elements:");
        for(i = 0; i <= num; i++) {
            scanf("%d", &arr[i]);
        }
        for(i = 0; i <= num; i++) {
            for(j = i + 1; j<= num; j++) {
                if(arr[i] == arr[j]) {
                    count = count + 1;
                }
            }
        }
        printf("Number of duplicate elements array are : %d\n",count);
        return 0;
        }
#endif



#if 0 //1
    int main()
    {
        int i;
        int arr[100];
        int arrodd[100];
```

```
int arreven[100];
int num;
int j = 0;
int k = 0;
printf("Enter the number of elements ;");
scanf("%d",&num);
printf("Enter array elements:");
for(i = 0; i < num; i++) {
	scanf("%d",&arr[i]);
}
for(i = 0; i < num; i++) {
	if(arr[i] % 2 == 0) {
		arreven[j] = arr[i];
		j++;
	} else {
		arrodd[k] = arr[i];
		k++;
		}
	}
	printf("Even elements in array are:");
	for(i = 0; i < j; i++) {
		printf("%d\n",arreven[i]);
	}

	printf("odd elements in array are:");
	for(i = 0; i < k; i++) {
		printf("%d\n",arrodd[i]);
	}
	return 0;
```

```
                }
#endif
```

```
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays$ cd arrsum/hdr/
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays/arrsum/hdr$ cat header.h
#include<stdio.h>
int sum(int arr[]);
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays/arrsum/hdr$ cd ../src/
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays/arrsum/src$ cat main.c
#include"header.h"
void main()
{
	int arr[5] = {1, 2, 3, 4, 5};
	int	result;
	result = sum(arr);
	printf("sum of array elements id %d \n", result);
}
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays/arrsum/src$ cat func.c
int sum(int arr[])
{
	int total = 0;
	int i;
	for(i = 0; i < 5; i++) {
		total = total + arr[i];
	}
	return total;
}
```

```
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays$ cd primearr/
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays/primearr$ ls
hdr  obj  prime  src
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays/primearr$ cd hdr/
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays/primearr/hdr$ cat header.h
#include<stdio.h>
int primenum(int num, int arr[]);
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays/primearr/hdr$ cd ../src/
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays/primearr/src$ cat main.c
#include"header.h"
void main()
{
	int num;
	int arr[50];
	int i;
	printf("enter the number of arrays elements :");
	scanf("%d",&num);
	printf("Enter array values:");
	for(i = 0; i <= num; i++) {
		scanf("%d",&arr[i]);
	}

	 primenum(num,arr);
}
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays/primearr/src$ cat func.c
#include"header.h"
int primenum(int num, int arr[])
{
```

```
	int i;
	int j;
	printf("prime numbers in array are:");
	for(i = 0; i <= num; i++) {
		int	count = 0;
		for(j = 1; j < arr[i]; j++) {
			if(arr[i] % j == 0) {
				count = count + 1;
			}
		}
				if(count < 2) {
					printf("%d\n",arr[i]);
				}
	}
	return 0;
}
```

```
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays$ cd seats/
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays/seats$ ls
hdr  obj  seatlayout  src
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays/seats$ cd hdr/
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays/seats/hdr$ cat header.h
#include<stdio.h>
int seats(int rows, int cols);
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays/seats/hdr$ cd ../
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays/seats$ cd src/
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays/seats/src$ cat main.c
```

```
#include"header.h"
int main()
{
	int rows;
	int cols;
	printf("Enter the number of rows:");
	scanf("%d", &rows);
	printf("Enter the number of cols:");
	scanf("%d", &cols);
	seats(rows,cols);
	return 0;
}
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/arrays/seats/src$ cat func.c
#include"header.h"
int seats(int rows, int cols)
{
	int arr[100][100];
	int i;
	int j;
	for(i = 1; i <= rows; i++) {
		for(j = 1; j <= cols; j++) {
			printf("	___%d",i);
		}
	printf("\n");
	}
	return 0;
}
```

```
srilatha@GESLMP22WP7T:~/Experiments/misc/practise$ cat endianess.c
#include<stdio.h>
#include<stdlib.h>

int     main(int argc, char **argv)
{

union {
        short s;
        char c[sizeof(short)];
        } un;
        un.s = 0x0102;
        //printf("%s: ", CPU_VENDOR_OS);
        if (sizeof(short) == 2) {
                if (un.c[0] == 1 && un.c[1] == 2)
                        printf("big-endian\n");
                else if (un.c[0] == 2 && un.c[1] == 1)
                        printf("little-endian\n");
                 else
                 printf("unknown\n");
         } else
                 printf("sizeof(short) = %d\n", sizeof(short));
exit(0);
}
srilatha@GESLMP22WP7T:~/Experiments/misc/practise$ cat ex
exam       example.c  exp/
srilatha@GESLMP22WP7T:~/Experiments/misc/practise$ cat example.c
#include<stdio.h>
```

```
#if QUES==4

int main()
{
	char str[10] = "srilatha";
	printf("%s",str);
return 0;
}

#endif



#if QUES==3
//not allowed
int main()
{
	char str[10];
	str = "srilatha";
	printf("%s",str);
return 0;
}

#endif
#if QUES==2

int main()
{
	char *str;
	str = "srilatha";
```

```
	printf("%s",str);
return 0;
}

#endif



#if QUES==1

int main()
{
	char *str = "srilatha";
	//printf("%c \n",str[1]);
	//str[1] = 'z';
	printf("%s",str);
	return 0;
}

#endif
```

#MALLOC

```
#include<stdio.h>
#include<stdlib.h>
#define MAXROW 3
#define MAXCOL 4

int main()
{
	int(*p)[MAXCOL];
	p = (int(*)[MAXCOL])malloc(MAXROW * sizeof(*p));
	printf("%d",p);
return 0;
}
```

****

srilatha@GESLMP22WP7T:~/Experiments/misc/practise/os_syscalls/myfile$ cat relation_btw_parentchild.c

```
#include<stdio.h>
#include<unistd.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/wait.h>
#include<stdlib.h>

int main(int argc, char *argv[])
{
    int fd, flags;
    char template[] = "/tmp/siriXXXXXX";
    setbuf(stdout, NULL);                                   /* Disable buffering of stdout */
    fd = mkstemp(template);
    if (fd == -1)
        exit(1);
    printf("File offset before fork(): %lld\n", (long long) lseek(fd, 0, SEEK_CUR));
    flags = fcntl(fd, F_GETFL);
    if (flags == -1)
        exit(1);
    printf("O_APPEND flag before fork() is: %s\n",(flags & O_APPEND) ? "on" : "off");
    switch (fork()) {
        case -1:
            exit(1);
        case 0:
            printf("child execution\n");
```

```
                if (lseek(fd, 1000, SEEK_SET) == -1)
                    exit(1);
                flags = fcntl(fd, F_GETFL);                                    /*
Fetch current flags */
                if (flags == -1)
                    exit(1);
                flags |= O_APPEND;
      /* Turn O_APPEND on */
                printf("%d",getpid());
                if (fcntl(fd, F_SETFL, flags) == -1)
                    exit(1);
                _exit(EXIT_SUCCESS);
            default:
                if (wait(NULL) == -1)
                    exit(1);
            /* Wait for child exit */
      printf("Child has exited\n");
      printf("%d",getpid());
      exit(0);
      printf("File offset in parent: %lld\n", (long long) lseek(fd, 0, SEEK_CUR));
      flags = fcntl(fd, F_GETFL);
      if (flags == -1)
            exit(1);
      printf("O_APPEND flag in parent is: %s\n", (flags & O_APPEND) ? "on" : "off");
            exit(EXIT_SUCCESS);
      }
}

/*#include<stdio.h>
```

```
#include<unistd.h>
#include<sys/types.h>
#include<sys/wait.h>
#include<stdlib.h>

pid_t child;
int main()
{
	printf("\nParent");
	int istack = 333;
	int idata = 111;
	int a = 10;
	printf("\nParent id : %d",getpid());
	switch(child = fork()) {
		case -1:
			printf("\nchild not created");
			break;
		case 0:
			//exit(0);
			printf("\nchild Helloooo");
			istack *= 3;
			a = a + 5;
			idata *= 2;
			//printf("\nchild id : %d",getpid());
			//printf("\nParent id : %d",getppid());
			//printf("\na = %d", a);
			break;
		default:
			//sleep(5);
```

```
                wait(NULL);
                printf("\n");
        }
        printf("\nistack = %d", istack);
        printf("\nidata  = %d", idata);
                printf("\na = %d", a);
        printf("\n\n");
return 0;
}
*/
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/os_syscalls/myfile$ cat syscalls.c
#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/wait.h>
#include<stdlib.h>

pid_t child;
int main()
{
    printf("\nParent");
    int istack = 333;
    int idata = 111;
    printf("\nParent id : %d",getpid());
    switch(child = fork()) {
        case -1:
            printf("\nchild not created");
            break;
        case 0:
```

```
            //exit(0);
            printf("\nchild Helloooo");
            istack *= 3;
            idata *= 2;
            printf("\nchild id : %d",getpid());
            printf("\nParent id : %d",getppid());
            break;
        default:
            //sleep(5);
            wait(NULL);
            printf("\n");
        }
        printf("\nistack = %d", istack);
        printf("\nidata  = %d", idata);
        printf("\n\n");
return 0;
}

/* here parent will execute first , there is drawback of executing the parent process first as it makes changes in child memory frame*/
/*#include<unistd.h>
#include<stdio.h>

pid_t child;
int main()
{
    printf("\nParent");
    printf("Parent id : %d",getpid());
    switch(child = fork()) {
```

```
        case -1:
            printf("\nchild not created");
            break;
        case 0:
            printf("\nchild Helloooo");
            printf("\n%d",getpid());
            printf("Parent id : %d",getppid());
            break;
        default:
            printf("\n99999Errorrrr");
        }
        printf("\n\n");
return 0;
}
*/


/* here child executes first as the parent process made to sleep till child process is being terminated */
/*#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/wait.h>
#include<stdlib.h>

pid_t child;
int main()
{
    printf("\nParent");
    printf("\nParent id : %d",getpid());
```

```
	switch(child = fork()) {
		case -1:
			printf("\nchild not created");
			break;
		case 0:
			//exit(0);
			printf("\nchild Helloooo");
			printf("\nchild id : %d",getpid());
			printf("\nParent id : %d",getppid());
			break;
		default:
			//sleep(5);
			wait(NULL);
			printf("\n99999Errorrrr");
		}
		printf("\n\n");
return 0;
} */

/*#include<stdio.h>
#include<unistd.h>
int main()
{
	printf("\nPID : %d", getpid());
	printf("\nPPID : %d", getppid());
	printf("\nfork return value : %d",fork());
	printf("\nPPID : %d", getppid());

}*/
```

srilatha@GESLMP22WP7T:~/Experiments/misc/practise/os_syscalls/myfile$ cat vfork.c

```
/* in fork by default parent process executes first,
whereas in vfork() child executes first and the data
changes also reflected in parent*/
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#include <stdlib.h>



static int idata = 10; /* Allocated in data segment */



int main(int argc, char *argv[])
{
      int istack = 20; /* Allocated in stack segment */
   pid_t childPid;
   switch (childPid = vfork()) {
      case -1:
            exit(1);
      case 0:
           idata *=
        istack *= 3;
        //execv("/usr/include/unistd.h",argv);
        break;
      default:
        sleep(0); /* Give child a chance to execute */
```

```
        //wait(NULL);
        break;
    }
    /* Both parent and child come here */
    printf("PID=%ld %s idata=%d istack=%d\n", (long) getpid(),(childPid == 0) ? "(child) " : "(parent)",
idata, istack);
    exit(EXIT_SUCCESS);
}
```

srilatha@GESLMP22WP7T:~/Experiments/misc/practise/os_syscalls/myfile$ cat relation_btw_parentchild.c

```
#include<stdio.h>
#include<unistd.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/wait.h>
#include<stdlib.h>

int main(int argc, char *argv[])
{
    int fd, flags;
    char template[] = "/tmp/siriXXXXXX";
    setbuf(stdout, NULL);                                                    /*
Disable buffering of stdout */
    fd = mkstemp(template);
    if (fd == -1)
        exit(1);
    printf("File offset before fork(): %lld\n", (long long) lseek(fd, 0, SEEK_CUR));
    flags = fcntl(fd, F_GETFL);
    if (flags == -1)
```

```
        exit(1);
    printf("O_APPEND flag before fork() is: %s\n",(flags & O_APPEND) ? "on" : "off");
    switch (fork()) {
        case -1:
            exit(1);
        case 0:
            printf("child execution\n");
            if (lseek(fd, 1000, SEEK_SET) == -1)
                exit(1);
            flags = fcntl(fd, F_GETFL);                                                    /*
Fetch current flags */
            if (flags == -1)
                exit(1);
            flags |= O_APPEND;
    /* Turn O_APPEND on */
            printf("%d",getpid());
            if (fcntl(fd, F_SETFL, flags) == -1)
                exit(1);
            _exit(EXIT_SUCCESS);
        default:
            if (wait(NULL) == -1)
                exit(1);
        /* Wait for child exit */
    printf("Child has exited\n");
    printf("%d",getpid());
    exit(0);
    printf("File offset in parent: %lld\n", (long long) lseek(fd, 0, SEEK_CUR));
    flags = fcntl(fd, F_GETFL);
    if (flags == -1)
```

```
        exit(1);
    printf("O_APPEND flag in parent is: %s\n", (flags & O_APPEND) ? "on" : "off");
        exit(EXIT_SUCCESS);
    }
}

/*#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/wait.h>
#include<stdlib.h>

pid_t child;
int main()
{
    printf("\nParent");
    int istack = 333;
    int idata = 111;
    int a = 10;
    printf("\nParent id : %d",getpid());
    switch(child = fork()) {
        case -1:
            printf("\nchild not created");
            break;
        case 0:
            //exit(0);
            printf("\nchild Helloooo");
            istack *= 3;
            a = a + 5;
```

```
                idata *= 2;
                //printf("\nchild id : %d",getpid());
                //printf("\nParent id : %d",getppid());
                //printf("\na = %d", a);
                break;
        default:
                //sleep(5);
                wait(NULL);
                printf("\n");
        }
        printf("\nistack = %d", istack);
        printf("\nidata  = %d", idata);
                printf("\na = %d", a);
        printf("\n\n");
return 0;
}
*/
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/os_syscalls/myfile$ cat syscalls.c
#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/wait.h>
#include<stdlib.h>

pid_t child;
int main()
{
    printf("\nParent");
    int istack = 333;
```

```
        int idata = 111;
        printf("\nParent id : %d",getpid());
        switch(child = fork()) {
                case -1:
                        printf("\nchild not created");
                        break;
                case 0:
                        //exit(0);
                        printf("\nchild Helloooo");
                        istack *= 3;
                        idata *= 2;
                        printf("\nchild id : %d",getpid());
                        printf("\nParent id : %d",getppid());
                        break;
                default:
                        //sleep(5);
                        wait(NULL);
                        printf("\n");
                }
                printf("\nistack = %d", istack);
                printf("\nidata  = %d", idata);
                printf("\n\n");
return 0;
}
```

/* here parent will execute first , there is drawback of executing the parent process first as it makes changes in child memory frame*/

```
/*#include<unistd.h>
#include<stdio.h>
```

```
pid_t child;
int main()
{
	printf("\nParent");
	printf("Parent id : %d",getpid());
	switch(child = fork()) {
		case -1:
			printf("\nchild not created");
			break;
		case 0:
			printf("\nchild Helloooo");
			printf("\n%d",getpid());
			printf("Parent id : %d",getppid());
			break;
		default:
			printf("\n99999Errorrrr");
		}
		printf("\n\n");
return 0;
}
*/


/* here child executes first as the parent process made to sleep till child process is being terminated */
/*#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/wait.h>
```

```
#include<stdlib.h>


pid_t child;
int main()
{
	printf("\nParent");
	printf("\nParent id : %d",getpid());
	switch(child = fork()) {
		case -1:
			printf("\nchild not created");
			break;
		case 0:
			//exit(0);
			printf("\nchild Helloooo");
			printf("\nchild id : %d",getpid());
			printf("\nParent id : %d",getppid());
			break;
		default:
			//sleep(5);
			wait(NULL);
			printf("\n99999Errorrrr");
		}
		printf("\n\n");
return 0;
} */

/*#include<stdio.h>
#include<unistd.h>
int main()
```

```
{
	printf("\nPID : %d", getpid());
	printf("\nPPID : %d", getppid());
	printf("\nfork return value : %d",fork());
	printf("\nPPID : %d", getppid());

}*/
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/os_syscalls/myfile$ cat vfork.c
/* in fork by default parent process executes first,
whereas in vfork() child executes first and the data
changes also reflected in parent*/
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#include <stdlib.h>



static int idata = 10; /* Allocated in data segment */



int main(int argc, char *argv[])
{
	int istack = 20; /* Allocated in stack segment */
   pid_t childPid;
   switch (childPid = vfork()) {
	case -1:
		exit(1);
```

```
        case 0:
             idata *=
          istack *= 3;
          //execv("/usr/include/unistd.h",argv);
          break;
        default:
          sleep(0); /* Give child a chance to execute */
          //wait(NULL);
          break;
        }
        /* Both parent and child come here */
        printf("PID=%ld %s idata=%d istack=%d\n", (long) getpid(),(childPid == 0) ? "(child) " : "(parent)",
idata, istack);
        exit(EXIT_SUCCESS);
}
```

****
****

srilatha@GESLMP22WP7T:~/Experiments/misc/practise$ cd patterns/

srilatha@GESLMP22WP7T:~/Experiments/misc/practise/patterns$ ls

alpha alphapatt.c charpat charpat.c hel.c numpat numpat.c out righttri.c squarepatt.c squarepattnum.c squpatt squpattnum starpatt starpatt.c

srilatha@GESLMP22WP7T:~/Experiments/misc/practise/patterns$ cat alphapatt.c

```
#include<stdio.h>
#if 1
    int main()
    {
        int i;
        int num;
        int j;
        printf("Enter the number of rows:");
        scanf("%d",&num);
        for(i = 0; i <= num; i++) {
            for(j = 0; j <= i; j++) {
                printf("%c", 65+i);
            }
            printf("\n");
        }
        return 0;
    }
#endif
```

srilatha@GESLMP22WP7T:~/Experiments/misc/practise/patterns$ cat charpat.c

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

```
int main()

{
	int row;
	int column;
	for(row = 1; row <= 5; row++) {
		for(column = 1; column <= 5; column++) {
			printf(" %c ",64+column);
		}
	printf("\n");
	}
return 0;
}
```

srilatha@GESLMP22WP7T:~/Experiments/misc/practise/patterns$ cat hel.c

```
#include<stdio.h>
int main()
{
printf("enter gvcgdrsd");
return 0;
}
```

srilatha@GESLMP22WP7T:~/Experiments/misc/practise/patterns$ cat numpat.c

```
#include<stdio.h>

int main()

{
```

```
        int row;
        int column;
        int key;
        for(row = 1; row <= 5; row++) {
                for(column = 1; column <= 5; column++) {
                        key=row+column;
                        if(key > 5) {
                                key = key - 5;
                                printf(" %d ",key);
                        } else {
                                printf(" %d ",key);
                        }

                }
        printf("\n");
        }
return 0;
}

srilatha@GESLMP22WP7T:~/Experiments/misc/practise/patterns$ cat righttri.c
#include<stdio.h>
int main()
{
        #if 1
                int row;
                int column;
                for(row = 1; row <= 5; row++) {
                        for(column = 1; column <= row; column++) {
                                printf(" * ");
```

```
            }
            printf("\n");
        }
        for(row = 1; row <= 1; row++) {
            for(int col = 1; col <= 6; col++) {
                printf(" * ");
            }
                printf("\n");
        }
        for(row = 1; row <= 5; row++) {
            for(column = 5; column >= row; column--) {
                printf(" * ");
            }
            printf("\n");
        }
    #endif
    #if 0
        int row;
        int column;
        for(row = 5; row <= 1; row++) {
        //     for(column = row; column >= 2*row-1; column++) {
                printf(" * ");
        //     }
            printf("\n");
        //     }
    }
    #endif
return 0;
}
```

srilatha@GESLMP22WP7T:~/Experiments/misc/practise/patterns$ cat squarepatt.c

```
/*      * * * * *
        * * * * *
        * * * * *
        * * * * *
        * * * * *
*/
```

```
#include<stdio.h>

int main()

{
    int row;
    int column;
    for(row = 1; row <= 5; row++) {
        for(column = 1; column <= 5; column++) {
            printf(" * ");
        }
    printf("\n");
    }
return 0;
}
```

```
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/patterns$ cat s
squarepatt.c    squarepattnum.c  squpatt        squpattnum      starpatt        starpatt.c
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/patterns$ cat squarepattnum.c

#include<stdio.h>

int main()

{
        int row;
        int column;
        for(row = 1; row <= 5; row++) {
                for(column = 1; column <= 5; column++) {
                        printf(" %d ",row);
                }
        printf("\n");
        }
return 0;
}

srilatha@GESLMP22WP7T:~/Experiments/misc/practise/patterns$ cat starpatt.c
#include<stdio.h>
int main()
{
#if 0
        int rows;
        int cols;
        int sp;
        int totalrows;
```

```
	int i = 0;
	printf("Enter the number rows : ");
	scanf("%d", &totalrows);
	for(rows = 1; rows <= 4; rows++) {
		for(sp = rows; sp <= 4; sp++) {
			printf("   ");
		}
		for(cols = 1; cols <= rows + i; cols++) {
			printf(" * ");
		}
		printf("\n");
		i++;
	}
#endif
#if 0 /* hallow square pattern*/
	int rows;
	int cols;
	int num;
	for(rows = 1;rows <= num; rows = rows + 1) {
	printf("enter number of rows :");
	scanf("%d", &num);
		if(rows == 1 || rows == 5) {
			for(cols = 1; cols <= num; cols++) {
				printf(" * ");
			}
		}
		else {
			printf(" * ");
				for(int space = 1; space < num - 1; space++) {
```

```
                printf("   ");
                }
            printf(" * ");
            }
        printf("\n");
    }
#endif
#if 0 /* c pattern*/
    int rows;
    int cols;
    int num;
    printf("enter number of rows :");
    scanf("%d", &num);
    for(rows = 1;rows <= num; rows = rows + 1) {
        if(rows == 1 || rows == 5) {
            for(cols = 1; cols <= num - 1; cols++) {
                printf(" * ");
            }
        }
        else {
            printf(" * ");
            }
        printf("\n");
    }
#endif
#if 0
    int rows;
    int cols;
    int num;
```

```
	int i=0;
	printf("Enter number of rows :");
	scanf("%d", &num);
	for(rows = 1; rows <= num; rows++) {
		for(int sp = rows;sp < num;sp++) {
			printf("   ");
		}
		for(cols = 1; cols <= rows + i; cols++) {
			printf(" * ");
		}
	i++;
	printf("\n");
	}
	int j=num-3;
	for(rows = num; rows >= 1; rows--) {
		for(int sp = num;sp >= rows;sp--) {
			printf("   ");
		}
		for(cols = 1; cols <= rows+j ; cols++) {
			printf(" * ");
		}
	j--;
	printf("\n");
}
#endif
#if 1
	int rows;
	int cols;
	int num;
```

```
	int i=0;
	printf("Enter number of rows :");
	scanf("%d", &num);
	for(rows = 1; rows <= num; rows++) {
//		for(int sp = rows;sp < num;sp++) {
//			printf("   ");
//		}
		if(rows == 1 || rows == num) {
		//for(cols = 1; cols <= rows + i; cols++) {
			printf(" * ");
		}
	i++;
	printf("\n");
	}
/*	int j;
	j = num - 3;
	for(rows = num; rows >= 1; rows--) {
		for(int sp = num;sp >= rows;sp--) {
			printf("   ");
		}
		for(cols = 1; cols <= rows+j ; cols++) {
			printf(" * ");
		}
	j--;
	printf("\n");
}*/
#endif
	return 0;
}
```

```
********************************************************************************
********************************************************************************
********************************************************************************
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/os_syscalls$ cd ../train1/
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/train1$ ls
train  train.c
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/train1$ cat train.c
#include<stdio.h>
int main()
{
	int array[50][50]; //to store 80 seats
	int rows;//to store for number of rows
	int cols;//to store for number of columns
	int seats;//to take input from users for number of seats
	int count = 0;
	int i=1;
	int select;
	printf("Enter the number of seats you want to book");
	scanf("%d",&seats);
	if(seats > 7) {
		printf("only 7 seats can be selected at once");
		for(rows = 1; rows <= 11; rows++) {
			printf("\n");
			for(cols = 1;cols <= 7;cols++) {
				printf("  %d ______  ",array[rows][cols]);
				i++;
			if(i > 80)
				break;
```

```
            }
        }
    }
    array[1][2]=1;
    array[1][4]=1;
    array[2][1]=1;
    array[3][2]=1;
    array[4][7]=1;
    array[5][1]=1;
    array[6][6]=1;
    array[7][4]=1;
    array[7][5]=1;
    array[8][8]=1;
    array[9][7]=1;
    array[7][2]=1;
    array[6][1]=1;
/*  for(rows = 1; rows <= 12; rows++) {
        for(cols = 1;cols <= 7;cols++) {
            if(array[rows][cols]!='X') {
            count = count+1;
                if(count == 7) {
                    printf("\nnearby seats are available in %d row\n", rows);
                    break;
                }
            }
        }
    }*/
    for(rows = 1; rows <= 12; rows++) {
```

```
        printf("\n");
        for(cols = 1;cols <= 7;cols++) {
            printf("   %d _____",array[rows][cols]);
            count=count+1;
            if(count == 80)
                break;
        }
    }
    for(rows = 1; rows <= 12; rows++) {
        for(cols = 1;cols <= 7;cols++) {
            if(array[rows][cols]!=1) {
                count = count+1;
                if(count >= seats)
                    printf("\nnearby seats are available in %d row\n", rows);
                  select = rows;
                break;
            }
                  select = rows;
                    break;
        }
        break;
    }
    printf("\nyour seats are booked in %d row",select);
    printf("\nseat numbers  booked for you in %d row are",select);
    for(cols = 1; cols <= 7; cols++) {
        if(array[select][cols] == 0) {
            //printf("%d",cols);
            count = count + 1;
        }
```

```
	printf("%d",cols);
	if(count > seats)
			break;
	}
return 0;
}
```

$$$

```
srilatha@GESLMP22WP7T:~/Experiments/misc/practise$ cd exp/
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/exp$ ls
arithmetic  arithmeticop.c  catp  catp.c  example  example.c  func  functions.c  structures  swapnum
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/exp$ cat arithmeticop.c
#include<stdio.h>
#if 1
	int main()
	{
		char ch1 = 'A';
		char ch2 = 'B';

	#if 1
			printf("%c\n",ch1 + ch2);
		#endif
		#if 0
			printf("%c\n",ch1 - ch2);
```

```
        #endif
        #if 0
            printf("%c\n",ch1 * ch2);
        #endif
        #if 0
            printf("%c\n",ch1 / ch2);
        #endif
        #if 0
            printf("%c\n",ch1 % ch2);
        #endif
        return 0;
    }
#endif
#if 0
    int main()
    {

        int a = 50;
        int b = 4;

        #if 1
            printf("\n %d", a + b);
        #endif
        #if 0
            printf("\n %d", a - b);
        #endif
        #if 0
            printf("\n %d", a * b);
        #endif
```

```
        #if 0
            printf("\n %d", a / b);
        #endif
        #if 0
            printf("\n %d",a % b);
        #endif
        return 0;
    }
#endif
#if 0
    int main()
    {

        float a = 2.5;
        float b = 5.0;

        #if 1
            printf("\n %f", a + b);
        #endif
        #if 0
            printf("\n %f", a - b);
        #endif
        #if 0
            printf("\n %f", a * b);
        #endif
        #if 0
            printf("\n %f", a / b);
        #endif
        #if 0
```

```
            printf("\n %f",a % b);
        #endif
        return 0;
    }
#endif
#if 0
    int main()
    {

        double a = 2.5;
        double b = 5.0;

        #if 1
            printf("\n %lf", a + b);
        #endif
        #if 0
            printf("\n %lf", a - b);
        #endif
        #if 0
            printf("\n %lf", a * b);
        #endif
        #if 0
            printf("\n %lf", a / b);
        #endif
        #if 0
            printf("\n %lf",a % b);
        #endif
        return 0;
    }
```

```
#endif
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/exp$ cat example.c
#include<stdio.h>
int a;
int a;

int main()
{
	int num;
	int num2;
	printf("Enter the number:");
	scanf("%d",&num);
	scanf("%d",&num2);
	printf("the number is %d %d",num, num2);
return 0;
}
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/exp$ cat catp.c
#include<stdio.h>

int main(int argc, char *argv[])
{
	FILE *fp;
	char ch;
	fp = fopen(argv[1],"r");
	ch = getc(fp);
	while(ch != EOF) {
		printf("%c",ch);
		ch = getc(fp);
	}
```

```
        return 0;
}
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/exp$ cat functions.c
#include<stdio.h>
int a = 10;
void func1();
void func2();
void func3();
void func4();
void func5();

int main()
{
        func1();
        func2();
        func3();
        func4();
        func4();
        func5();
return 0;
}
void func1()
{
    extern int d;
        printf("function 1\n");
        printf("a is %d\n",a);
        printf("d is %d\n",d);
}
void func2()
```

```
{
	extern int d;
	printf("function 2\n");
	printf("a is %d\n",a);
	printf("d is %d\n",d);
}
void func3()
{
   extern int b;
	printf("function 3\n");
	printf("a is %d\n",a);
	printf("b is %d\n",b);
}
void func4()
{
	extern int b;
	printf("function 4\n");
	printf("a is %d\n",a);
	printf("b is %d\n",b);
}
void func5()
{
	extern int c;
	printf("function 5\n");
	printf("a is %d\n",a);
	printf("c is %d\n",c);
}
	int b = 20;
	int c = 30;
```

```
        int d = 40;
```

```
??????????????????????????????????????????????????????????????????????????????????????????????????
??????????????????????????????????????????????????????????????????????????????????????????????????
srilatha@GESLMP22WP7T:~/Experiments/misc/practise/exp/structures$ cat structure.c
#include<stdio.h>
#include<string.h>
#include<stdlib.h>



#if QUES ==3


int str_len(char*);


int main()
{       int length;
        char str[10] = "sri";
        length = str_len(str);
        printf("length of string is %d\n", length);
        return 0;
}
int str_len(char *str)
{
```

```
	int count = 0;
	while(*str++) {
		count+=1;
	}
	return count;
}
#endif




#if QUES ==2
struct stu{
	int roll;
	int x;
	int *p;
	}s1;
int main()
{
	struct stu s1, *ptr;
	ptr = &s1;
	s1.roll = 10;
	s1.x = 5;

	printf("roll is %d\n",(*ptr).roll);
	printf("roll is %d\n",ptr->roll);
	printf("ptr is %u\n",ptr);
return 0;
}
```

```
#endif



#if QUES ==1
int main()
{
	char arr[20];
	char *p;
	fgets(arr,20,stdin);
	p = (char *)malloc(sizeof(char));
	arr[2]='$';
//	 for(int i = 0; i <= 20; i++) {
		printf("%s",arr);
		printf("%u",p);
//	 }
return 0;
}
#endif
#if 0
int main()
{
	int a = 10;
	char b ='s';
	int *p;
	char *c;
	p = &a;
	c = &b;
	printf("\n p is %u",p);
```

```
	printf(" \nc is %s",c);
	printf(" \n*p is %d ",*p);
	printf(" \nc is %d",*c);
	printf("\npointer sum = %u\n",p+1);
	printf("\npointer subraction = %u\n",p-1);
//	 printf("\npointer multiplication = %u\n",p*1);
//	 printf("\npointer division = %u\n",p\1);
	printf("\npointer of char addition  = %d\n",c+1);
	return 0;
}
#endif

#if 0
struct stu{
	int a;
	int b;
	};
int main()
{
//	 struct stu *ptr;
	struct stu s1 = {1 , 2},*ptr;
	ptr = &s1.a;
	printf("s1 a is %d\n",s1.a);
	printf("s1 b using ptr is %d\n",ptr->b);
	printf("Address of s1 is %d",ptr);
	return 0;
}
#endif
#if 0
```

```
struct st1{
	int as1;
	struct st2{
		int as2;
	} s2;
} s1;
int main()
{
	printf("structure 1 %d\n",s1.as1);
	printf("Nested structure%d\n",s1.s2.as2);
return 0;
}
#endif
#if 0
struct stu{
	int roll;
	char name[10];
	int marks[3];
	};
	int main()
	{
		struct stu s1[5];
		int i;
		for(i = 1; i < =3; i++)
		{
			printf("Enter roll of student : ");
			scanf("%d",&s1[i].roll);
			printf("Enter name of student:");
			scanf("%s",s1[i].name);
```

```
            printf("Enter the marks of student");
            for(int k = 1; k <= 3; k++)
                scanf("%d",&s1[i].marks[k]);
        }
        for(i = 1; i < 3; i++)
        {
            printf("Student %d:\n",i);
            for(int j = 1; j < i+1; j++)
            {
                printf("Roll  :  %d\n",s1[i].roll);
                printf("Name  :  %s\n",s1[i].name);
                for(int k = 1; k <= 3; k++)
                {
                    printf("Marks in subject %d is %d",k,s1[i].marks[k]);
                    printf("Marks :  %d\n",s1[i].marks[k]);
                }
            }
        }
    return 0;
}
#endif
#if 0
struct emp{
    int id;
    char name[10];
    int age;
    };
    int main()
    {
```

```
        struct emp e1[5];
        int i;
        for(i = 1; i <5; i++)
        {
            printf("enter id of employee : ");
            scanf("%d",&e1[i].id);
            printf("Enter name of employe:");
            scanf("%s",e1[i].name);
            printf("Enter the age of employee");
            scanf("%d",&e1[i].age);
        }
        for(i = 1; i <= 5; i++)
        {
            printf("Employee %d:\n",i);
            for(int j = 1; j < i+1; j++)
            {
                printf("id  :  %d\n",e1[i].id);
                printf("Name:  %s\n",e1[i].name);
                printf("Age :  %d\n",e1[i].age);
            }
        }
    return 0;
}
#endif
#if 0
int main()
{
    struct student{
        int a;
```

```
        int b;
        int c;
        };
    struct student s1,s2;
    printf("Address of s1.student a is %d\n",&s1.a);
    printf("Address of s1.student b is %d\n",&s1.b);
    printf("Address of s1.student c is %d\n",&s1.c);
    printf("Address of s2.student a is %d\n",&s2.a);
    printf("Address of s2.student b is %d\n",&s2.b);
    printf("Address of s2.student c is %d\n",&s2.c);
    return 0;
}
#endif
#if 0
int main()
{
    int x;
    struct stu{
        int a;
        char c;
        float f;
        char d;
        };
    struct name{
        char name[10];
        };
    struct num{
        int a;
        };
```

```
    struct floa{
        float p;
        };

    struct arr{
        int num[10];
        };
    struct stu s1;
    x = sizeof(s1);
    int y = sizeof(struct name);
    printf("structure size is %ld\n",sizeof(struct stu));
    printf("structure variable size is %d\n",x);
    printf("structure name size is %d\n",y);
    printf("structure number size is %d\n",sizeof(struct num));
    printf("structure float  size is %d\n",sizeof(struct floa));
    printf("structure arr  size is %d\n",sizeof(struct arr));
    printf("structure last char size is %d\n",sizeof(struct stu));
return 0;
}
#endif
#if 0
int main()
{

    struct var{
        int a;
        int b;
        int c;
        }s={.b = 5, .c = 9};
```

```
	printf("%d %d\n",s.b,s.c);
return 0;
}
#endif
#if 0
int main()
{
	struct var{
		int a;
		int b;
		int c;
		};
	struct var s1 = {1,2,3,};
	struct var s2,s3;
	//var.s1 = {1,2,3};/* INVALID */
	s2.a = 7;
	s2.b = 8;
	s2.c = 9;
	s3 = s1;
	printf("s1 is %d %d %d\n", s1.a, s1.b, s1.c);
	printf("s2 is %d %d %d\n", s2.a, s2.b, s2.c);
	printf("s3 is %d %d %d\n", s3.a, s3.b, s3.c);
return 0;
}
#endif
#if 0
int main()
{
	struct var{
```

```
        char name[20];
        int b;
        int c;
        };
    struct var s1 = {"sri",2,3,};
    struct var s2,s3;
    //var.s1 = {1,2,3};/* INVALID */
    strcpy(s2.name, "latha");
    s2.b = 8;
    s2.c = 9;
    s3 = s1;
    printf("s1 is %s %d %d\n", s1.name, s1.b, s1.c);
    printf("s2 is %s %d %d\n", s2.name, s2.b, s2.c);
    printf("s3 is %s %d %d\n", s3.name, s3.b, s3.c);
return 0;
}
#endif
#if 0
#endif
#if 0
#endif
#if 0
#endif
```

0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

srilatha@GESLMP22WP7T:~/Experiments/misc/practise/exp/swapnum$ cat main.c

```
#include<stdio.h>


void swap(int, int);


int main()
{
	int a = 10;
	int b = 20;
	swap(a, b);
	return 0;
}
```

srilatha@GESLMP22WP7T:~/Experiments/misc/practise/exp/swapnum$ cat swap.c

```
#include<stdio.h>
extern int a[];
int *a1 = &a[0];
int *a2 = &a[1] ;
//extern int *a2;
/*void swap()
{
	a[0] = a [0] + a[1];
	a[1] = a[0] - a[1];
	a[0] = a[0] - a[1];
	//a = a + b;
	//b = a - b;
	//a = a - b;
	printf("a = %d",a[0]);
```

```
	printf("b = %d",a[1]); */



void swap()
{

	//extern int *a = &a;
	//extern int *a2 = *(a+1);
	*a1 = *a1 + *a2;
	*a2 = *a1 - *a2;
	*a1 = *a1 - *a2;



	printf("a = %d\n",*a1);
	printf("b = %d",*a2);
	printf("\n");
}
```
