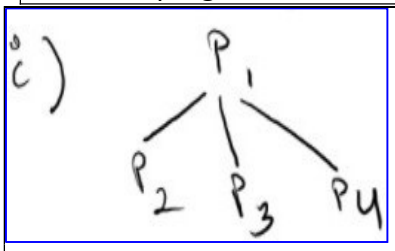
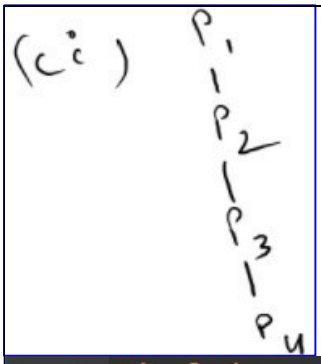
1.) Write a program to create the following process tree.



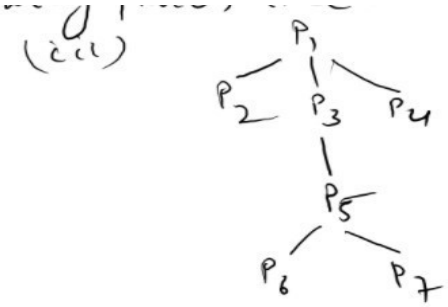

```c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int i, j, k;
    for(i=0; i<3; i++)
    {
        j=fork();
        if(j==0)
        break;
    }
    printf("%d %d\n",getpid(), getppid());
    while(k = wait(NULL)!= -1);

    return 0;
}
```

```
3171 3170
3175 3171
3177 3171
3176 3171


...Program finished with exit code 0
Press ENTER to exit console.
```




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

int main()
{
    int i, j, k;
    for(i=0; i<3; i++)
    {
        j=fork();
        if(j !=0)
        break;
    }
    printf("%d %d\n",getpid(), getppid());
    while(k = wait(NULL)!= -1);

    return 0;
}

```

```
4346 4345
4350 4346
4351 4350
4352 4351


...Program finished with exit code 0
Press ENTER to exit console.
```

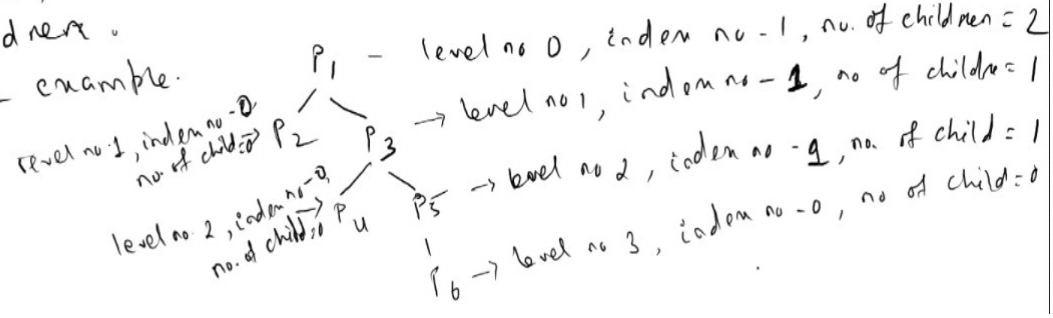



```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

int main()
{
   int i, j, k, l,m;
   for(i=0; i<3; i++)
   {
      j=fork();
      if(j==0)
      {
         if(i==1)
         {
            m=fork(); if(m!=0)
            break; for(l=0; l<2;
            l++)
            {
               m=fork(); if(m==0)
               break;
            }
         }
         break;
      }
   }

   printf("%d %d\n",getpid(), getppid());
   while(k = wait(NULL)!= -1);

   return 0;
}
```

```
                    ^~~~
2023 2019
2019 2018
2025 2019
2024 2019
2026 2024
2028 2026
2027 2026


...Program finished with exit code 0
Press ENTER to exit console.
```

(2) Write a c Program, that will create any given process tree with the following information for each process i.e. level number, index number and no. of children.

For example.




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

int main()
{
   int i, j, k, l,m, n;
   for(i=0; i<2; i++)
   {
      j=fork();
      if(j==0)
      {
         if(i==1)
         {
            for(l=0; l<2; l++)
            {


               m=fork(); if(m==0
               && l==0)
               {
                  break;
               }
               else if(m==0 && l==1)
               {
                  m=fork(); if(m==0)
                  break;
               }
            }
         }
         break;

      }
   }
   printf("%d %d\n",getpid(), getppid());
   while(k = wait(NULL)!= -1);
```

```
   return 0;
}
```

```
1533 1532
1538 1533
1537 1533
1540 1538
1539 1538
1541 1540


...Program finished with exit code 0
Press ENTER to exit console.
```

3.) Write a program to find sum of even numbers in parent process and sum of odd numbers inchild process.

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

int main()
{
      int i, t, s, n, sump = 0, sumc = 0;
      printf("Enter the value of n: ");
      scanf("%d",&n);
      int arr[n];
      printf("Enter the numbers:\n");
         for(i=0; i<n; i++)
         scanf("%d",&arr[i]);
         t = fork(); for(i=0;
         i<n; i++)
         {
         if(t > 0 && arr[i]%2 == 0)
         sump+=arr[i];
         if(t == 0 && arr[i]%2 != 0)
         sumc+=arr[i];
          }
       if(t>0)
      printf("Sum of even numbers by parent process= %d\n",
      sump);
      else
      printf("Sum of odd numbers by child process= %d\n", sumc);while(s = wait(NULL) != -1);
      return 0;
}
```

```
Enter the value of n: 7
Enter the numbers:
34
56
78
1
4
5
7
Sum of even numbers by parent process= 172
Sum of odd numbers by child process= 13
```