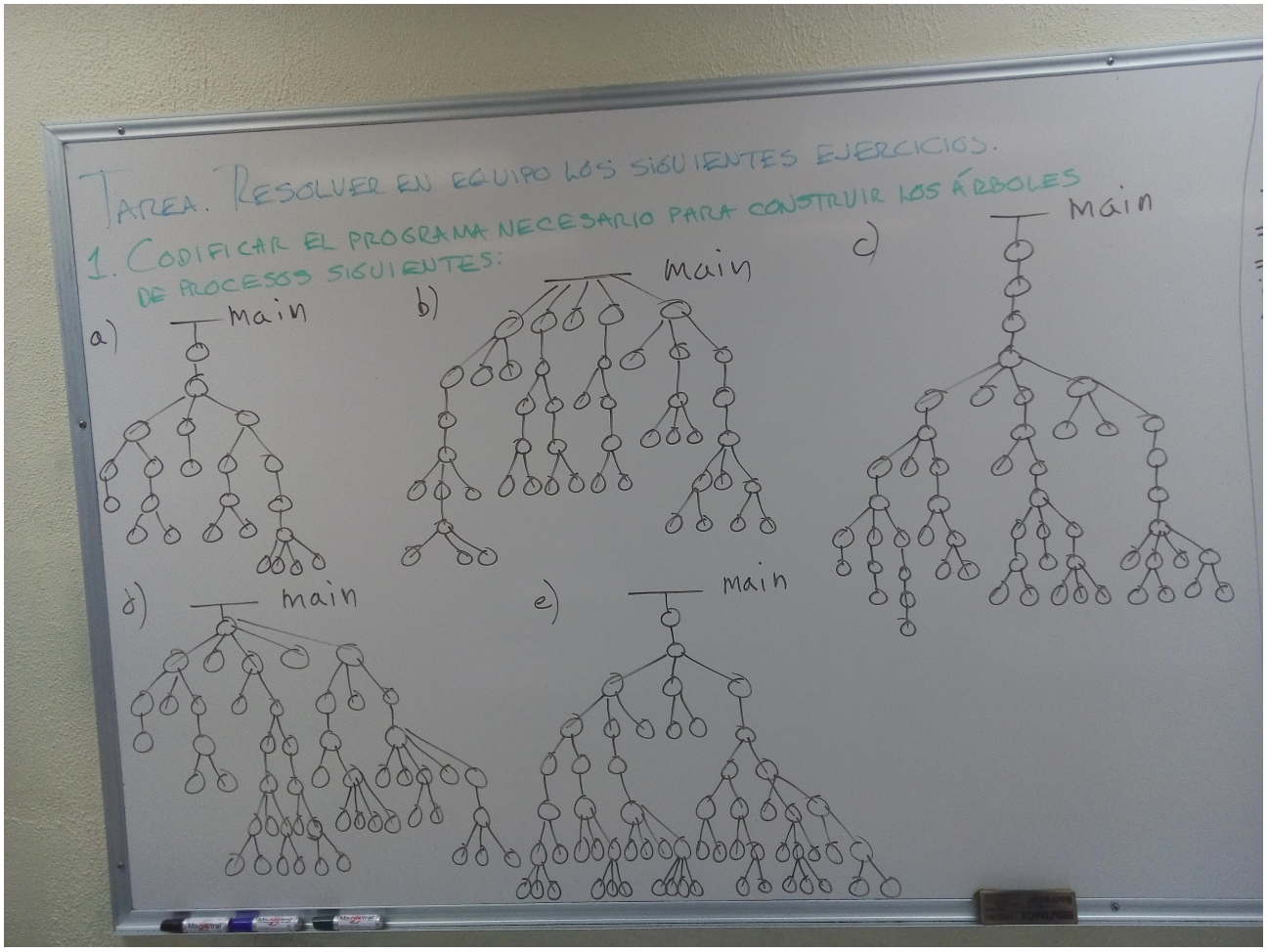
1. Codificar el programa necesario para construir los árboles de procesos siguientes:



2. Árbol A)

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

int main(void){
   int id_proc, i, j, k, l, m, w, x, y;

   for(i=0; i<2; i++){//2 procesos
      if((id_proc=fork())==0){//Creo 2 procesos verticales
         printf("Soy el proceso hijo\n");

         if(i==1){//Segundo arbol vertical
            for(j=0; j<3; j++){//3 procesos
               if((id_proc=fork())==0){//Creo 3 procesos horizontales
                  printf("Soy el proceso hijo\n");

                  if(j==0){//Primer proceso - izquierda
```

```
18                      for(k=0; k<2; k++){//2 procesos
19                         if((id_proc=fork())==0){//Creo 2 procesos horizontales
20                            printf("Soy el proceso hijo\n");
21
22                            if(k==0){//Primer proceso - izquierda
23                               if((id_proc=fork())==0){//1 proceso vertical
24                                  printf("Soy el proceso hijo\n");
25                                  exit(0);
26                               }//ACABAMOS
27                            }
28                            else if(k==1){//Segundo proceso - derecha
29                               if((id_proc=fork())==0){//1 proceso vertical
30                                  printf("Soy el proceso hijo\n");
31
32                                  for(l=0; l<2; l++){//2 procesos
33                                     if((id_proc=fork())==0){//Creamos 2 procesos horizontales
34                                        printf("Soy el proceso hijo\n");
35                                        exit(0);
36                                     }//ACABAMOS
37                                  }
38                                  exit(0);
39                               }
40                            }
41                            exit(0);
42                         }
43                      }
44                   }
45                   else if(j==1){//Segundo proceso - centro
46                      if((id_proc=fork())==0){//1 proceso vertical
47                         printf("Soy el proceso hijo\n");
48                         exit(0);
49                      }//ACABAMOS
50
51                   }
52                   else if(j==2){//Tercer proceso - derecha
53                      for(m=0; m<2; m++){//2 procesos
54                         if((id_proc=fork())==0){//Creo 2 procesos horizontales
55                            printf("Soy el proceso hijo\n");
56
57                            if(m==0){//Primer proceso - izquierda
58                               if((id_proc=fork())==0){//1 proceso vertical
59                                  printf("Soy el proceso hijo\n");
60
61                                  for(l=0; l<2; l++){//2 procesos
62                                     if((id_proc=fork())==0){//Creo 2 procesos horizontales
63                                        printf("Soy el proceso hijo\n");
64                                        exit(0);
65                                     }//ACABAMOS
66                                  }
67                                  exit(0);
68                               }
69                            }
70                            else if(m==1){//Segundo proceso - derecha
71                               for(w=0; w<2; w++){//2 procesos
72                                  if((id_proc=fork())==0){//Creamos 2 procesos verticales
73                                     printf("Soy el proceso hijo\n");
74                                     if(w==1){//Segundo proceso
75                                        for(x=0; x<4; x++){//4 procesos
76                                           if((id_proc=fork())==0){//Creamos 4 procesos horizontales
77                                              printf("Soy el proceso hijo\n");
78                                              exit(0);
79                                           }//ACABAMOS
80                                        }
81                                     }
82                                  }
83                                  else{
84                                     break;
85                                     exit(0);
86                                  }
```

```
                                    }
                                }
                                exit(0);
                            }
                        }
                    }
                    exit(0);
                }
            }
        }
    }
    else{
        break;
        exit(0);
    }
}
exit(0);
}
```

3. Árbol B)

```
// 04 de Octubre 2018
// Para generar muchos procesos HIJO
#include<unistd.h>
#include<stdio.h>
#include<stdlib.h>
#include <sys/types.h>
int i = 10;

void hijosVertical(int n)
{
    if (n == 0)
        exit(0);
    printf("proceso padre: %d\n", getppid());
    printf("proceso actual %d\n", getpid());
    int id_proc = fork();
    if (id_proc == 0)
    {
        printf("proceso hijo %d\n", getpid());
        nodosEnColados(--n);
        exit(0);
    }
    wait(0);
}

void hijosHorizontal(int n) {
    printf("proceso actual %d\n", getpid());
    int id_proc=0;
    for (int i = 0; i < n; i++)
    {
        if ((id_proc = fork()) == 0)
        {
            printf("proceso padre: %d\n", getppid());
            printf("proceso hijo: %d\n", getpid());
            exit(0);
        }
        wait(0);
    }
}
int main(void)
{
    int id_proc;
    for(i = 0; i<5; i++) // CREAR 5 PROCESOS HORIZONTALES
    {
        id_proc = fork();
```

```
45      if(id_proc == 0)
46      {
47         printf("Soy el proceso hijo %i\n", i);
48         if(i == 0)   // PARA EL PRIMER HIJO
49         {
50            for(j = 0; j<3; j++) // CREAR 3 PROCESOS HORIZONTALES
51            {
52               id_proc = fork();
53               if(id_proc == 0)
54               {
55                  printf("Soy el proceso hijo\n");
56                  if(j == 0) // PARA EL PRIMER HIJO
57                  {
58                     for(k = 0; k<2; k++) // CREA 2 PROCESOS VERTICALES
59                     {
60                        id_proc = fork();
61                        if(id_proc == 0)
62                        {
63                           printf("Soy el proceso hijo %i\n", i);
64                           if(k == 1) // EN EL ULTIMO PROCESO
65                           {
66                              for(l = 0; l<3; l++) // CREA  PROCESOS HORIZONTALES
67                              {
68                                 id_proc = fork();
69                                 if(id_proc == 0)
70                                 {
71                                    if(l == 1)
72                                    {
73
74                                       id_proc = fork();
75                                       if(id_proc == 0)
76                                       {
77                                          hijosHorizontal(3);
78                                       }
79                                    }
80                                 }
81                              }
82                              exit(0);
83                           }
84                        }
85                        else
86                        {
87                           break; // Sale del ciclo
88                        }
89                     }
90                     exit(0);
91                  } // Fin if - primer hijo
92               }
93               exit(0);
94            }
95            exit(0);
96         } // Fin if - Hijo 0
97
98         if(i == 1) // PARA EL SEGUNDO HIJO
99         {
100           id_proc = fork(); // Crear un hijo
101           if(id_proc == 0)
102           {
103              for(j = 0; j<2; j++) // Crear dos hijos horizontales
104              {
105                 id_proc = fork();
106                 if(id_proc == 0)
107                 {
108                    if(j == 0) // Para el primer hijo
109                    {
110                       id_proc = fork();
111                       if(id_proc == 0) // Crear un hijo
112                       {
113                          hijosHorizontal(2);
```

```
                        }
                    }
                    if(j == 1) // Para el segundo hijo
                    {
                        id_proc = fork();
                        if(id_proc == 0) // Crear un hijo
                        {
                            hijosHorizontal(2);
                        }
                    }
                }
            }
            exit(0);
        }
    }
    if(j == 2) // PARA EL TERCER HIJO
    {
        exit(0); // No tiene hijos
    }
    if(j == 3) // PARA EL CUARTO HIJO
    {
        id_proc = fork();
        if(id_proc == 0) // Crear un hijo
        {
            for(k = 0; k<2; k++) // Crear dos hijos
            {
                id_proc = fork();
                if(id_proc == 0)
                {
                    if(k == 0) // Para el hijo 1
                    {
                        exit(0);
                    }
                    if (k == 1) // Para el hijo 2
                    {
                        id_proc = fork();
                        if(id_proc == 0)
                        {
                            hijosHorizontal(2);
                        }
                    }
                }
            }
            exit(0);
        }
    }
    if(j == 4)
    {
        for(k = 0; k<3; k++) // Crear 3 hijos
        {
            id_proc = fork();
            if(id_proc == 0)
            {
                if(k == 0) // Para el hijo 1
                {
                    exit(0);
                }
                if(k == 1) // Para el hijo 2
                {
                    id_proc = fork();
                    if(id_proc == 0) // Crear un hijo
                    {
                        hijosHorizontal(3);
                    }
                }
                if(k == 2) // Para el hijo 3
                {
                    for(l = 0; l<2; l++) // Crea 2 hijos verticales
                    {
```

```
                        id_proc = fork();
                        if(id_proc == 0)
                        {
                            if(l == 1)
                            {
                                for(m = 0; m<3; m++)
                                {
                                    id_proc = fork();
                                    if(id_proc == 0)
                                    {
                                        if(m == 0 || m == 2)
                                        {
                                            hijosHorizontal(2);
                                        }
                                        if(m == 1)
                                        {
                                            exit(0);
                                        }
                                    }
                                }
                            }
                        }
                        else
                        {
                            break;
                        }
                    }

                }
            }
            exit(0);
        }
    }
}   // Fin 5 procesos horizontales
    exit(0);
}
```

4. Árbol C)

```
//Pal jueves 12-Oct-2018
#include <stdio.h>
#include <stdlib.h>

int main(void){
    int id_proc, i,j,k,l,m,p;
    for(i=0;i<4;i++)
    {
        id_proc=fork();//1
        if(id_proc==0){
            printf("\nSoy un proceso hijo")
            if(i==3){
                for(j=0;j<4;j++){
                    id_proc=fork();//2
                    if(id_proc==0){
                        if(j==0){//2.1
                            id_proc=fork();
                            if(id_proc==0){
                                for(k=0;k<3;k++){
                                    id_proc=fork();
                                    if(id_proc==0){
                                        printf("\nSoy un proceso hijo");
                                        if(k==0){//2.1.1
                                            id_proc=fork();
                                            if(id_proc==0){
```

```
26                          for(l=0;l<3;l++){
27                             id_proc=fork();
28                             if(id_proc==0){
29                                printf("\nSoy un proceso hijo");
30                                if(l==0){//2.1.1.1
31                                   id_proc=fork();
32                                   if(id_proc==0){
33                                      printf("\nSoy un proceso hijo");
34                                      exit(0);
35                                   }
36
37                                }
38                                else if(l==1){//2.1.1.2
39                                   printf("\nSoy un proceso hijo");
40                                   for(m=0,m<2;m++){
41                                      id_proc=fork();
42                                      if(id_proc==0){
43                                         printf("\nSoy un proceso hijo");
44                                          //exit(0);
45                                      }
46                                      else{
47                                         break;
48                                      }
49                                   }
50                                   exit(0);
51                                }
52                                else if(l==2){//2.1.1.3
53                                   printf("\nSoy un proceso hijo");
54                                   for(m=0,m<3;m++){
55                                      id_proc=fork();
56                                      if(id_proc==0){
57                                         printf("\nSoy un proceso hijo");
58                                          exit(0);
59                                      }
60                                      else{
61                                         printf("\nSoy un proceso padre");
62                                          break;
63                                      }
64                                   }
65                                   exit(0);
66                                }
67                                exit(0);
68                             }
69                             /*else{
70                                printf("\nSoy un proceso padre");
71                                //exit(0);
72                             }*/
73                          }
74                       }
75                       /*else{
76                          printf("\nSoy un proceso padre");
77                          //exit(0);
78                       }*/
79                    }
80                    else if(k==1){//2.1.2
81                          printf("\nSoy un proceso hijo");
82                          exit(0);
83                    }
84                    else if(k==2){//2.1.3
85                       id_proc=fork();
86                       if(id_proc==0){
87                          printf("\nSoy un proceso hijo");
88                          for(l=0;l<2;l++){
89                             id_proc=fork();
90                             if(id_proc==0){
91                                if(l==0){//2.1.3.1
92                                   printf("\nSoy un proceso hijo");
93                                   exit(0);
94                                }
```

```
                                    else if(l==1){//2.1.3.2
                                        printf("\nSoy un proceso hijo");
                                        for(m=0;m<2;m++){
                                            id_proc=fork();
                                            if(id_proc==0){
                                                printf("\nSoy un proceso hijo");
                                                exit(0);
                                            }
                                            /*else{
                                                printf("\nSoy un proceso padre");
                                            }*/
                                        }
                                        exit(0);
                                    }
                                }
                                /*else{
                                    printf("\nSoy un proceso padre");
                                    //exit(0);
                                }*/
                            }
                            exit(0);
                        }
                        exit(0);
                        /*else{
                            printf("\nSoy un proceso padre");
                            //exit(0);
                        }*/
                    }
                }
                /*else{
                    printf("\nSoy un proceso padre");
                    //exit(0);
                }*/
            }
        }
        /*else{
            printf("\nSoy un proceso padre");
            //exit(0);
        }*/
    }
    else if(j==1){//2.2
        printf("\nSoy un proceso hijo");
        exit(0);
    }
    else if(j==2){//2.3
        id_proc=fork();
        if(id_proc==0){
            printf("\nSoy un proceso hijo");
            for(k=0;k<2;k++){
                id_proc=fork();
                if(id_proc==0){
                    if(k==0){//2.3.1
                        printf("\nSoy un proceso hijo");
                        exit(0);
                    }
                    else if(k==1){//2.3.2
                        id_proc=fork();
                        if(id_proc==0){
                            printf("\nSoy un proceso hijo");
                            for(l=0;l<3;l++){
                                id_proc=fork();
                                if(id_proc==0){
                                    printf("\nSoy un proceso hijo");
                                    if(l==0){//2.3.2.1
                                        id_proc=fork();
                                        if(id_proc==0){
                                            printf("\nSoy un proceso hijo");
                                            for(m=0;m<2;m++){
                                                id_proc=fork();
```

```
                                    if(id_proc==0){
                                        printf("\nSoy un proceso hijo");
                                        exit(0);
                                    }
                                    /*else{
                                        printf("\nSoy un proceso padre");
                                    }*/
                                }
                            }
                            exit(0);
                            /*else{
                                printf("\nSoy un proceso padre");
                                //exit(0);
                            }*/

                        }
                        else if(l==1){//2.3.2.2
                            id_proc=fork();
                            if(id_proc==0){
                                printf("\nSoy un proceso hijo");
                                exit(0);
                            }
                            /*else{
                                printf("\nSoy un proceso padre");
                                //exit(0);
                            }*/

                        }
                        else if(l==2){//2.3.2.3
                            id_proc=fork();
                            if(id_proc==0){
                                printf("\nSoy un proceso hijo");
                                for(m=0;m<3;m++){
                                    id_proc=fork();
                                    if(id_proc==0){
                                        printf("\nSoy un proceso hijo");
                                        exit(0);
                                    }
                                    /*else{
                                        printf("\nSoy un proceso padre");
                                        //exit(0);
                                    }*/
                                }
                            }
                            exit(0);
                            /*else{
                                printf("\nSoy un proceso padre");
                                //exit(0);
                            }*/
                        }
                    }
                    /*else{
                        printf("\nSoy un proceso padre");
                        //exit(0);
                    }*/
                }
            }
            exit(0);
            /*else{
                printf("\nSoy un proceso padre");
                //exit(0);
            }*/

        }
    }
    exit(0);
    /*else{
        printf("\nSoy un proceso padre");
        //exit(0);
```

```
                        }*/
                    }
                }
                exit(0);
                /*else{
                    printf("\nSoy un proceso padre");
                    //exit(0);
                }*/
            }
            else if(j==3){//2.4
                for(p=0;p<3;p++){
                    id_proc=fork();
                    if(id_proc==0){
                        if(p<2){//2.4.1 y 2.4.2
                            printf("\nSoy un proceso hijo");
                            exit(0);
                        }
                        if(p==2){//2.4.3
                            printf("\nSoy un proceso hijo");
                            for(k=0;k<3;k++){
                                id_proc=fork();
                                if(id_proc==0){
                                    printf("\nSoy un proceso hijo");
                                    if(k==3){
                                        for(l=0;l<4;l++){
                                            id_proc=fork();
                                            if(id_proc==0){
                                                if(l==0 || l==2){//2.4.3.1 y 2.4.3.3
                                                    printf("\nSoy un proceso hijo");
                                                    exit(0);
                                                }
                                                else if(l==1 || l==3){//2.4.3.2 y 2.4.3.4
                                                    printf("\nSoy un proceso hijo");
                                                    for(m=0;m<2;m++){
                                                        id_proc=fork();
                                                        if(id_proc==0){
                                                            printf("\nSoy un proceso hijo");
                                                            exit(0);
                                                        }
                                                        /*else{
                                                            printf("\nSoy un proceso padre");
                                                            //exit(0);
                                                        }*/
                                                    }
                                                    exit(0);
                                                }
                                            }
                                            exit(0);
                                            /*else{
                                                printf("\nSoy un proceso padre");
                                                //exit(0);
                                            }*/
                                        }

                                    }
                                }
                                exit(0);
                                /*else{
                                    printf("\nSoy un proceso padre");
                                    break;
                                }*/
                            }

                        }
                    }
                    exit(0);
                    /*else{
                        printf("\nSoy un proceso padre");
                        //exit(0);
```

```
                        } */
                    }

                }
            }
            exit(0);
            /*else{
               printf("\nSoy un proceso padre");
               //exit(0);
            }*/
        }
    }
    else{
       break;
       exit(0);
    }
  }
  exit(0);
}
```

5. Árbol D)

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

int main(int argc, char const *argv[])
{
   int id_proc, index = 0;
   if (fork() == 0)
   {
      exit(0);
   }else{
      // printf("First fork\n");
      for (int i = 0; i < 3; ++i)
      {
         if(fork() == 0){
            index += pow(2,i);
         }
         if (index > 4)
         {
            exit(0);
         }
      }
      // printf("%i\n",index);
      int aux1 = 0;
      if (index == 0 || index == 4)
      {
         for (int i = 0; i < 2; ++i)
         {
            if(fork() == 0){
               aux1 += pow(2,i);
            }
            if (aux1 > 2)
            {
               exit(0);
            }
         }
         // printf("%i:%i\n",index,aux1);
         int aux2 = 0;
         if (aux1 == 0 || aux1 == 2)
         {
            if (fork() == 0)
```

```
43              {
44                 exit(0);
45              }
46              // printf("%i:%i:%i\n",index,aux1,aux2);
47              int aux3 = 0;
48              if (index == 0 && aux1 == 2)
49              {
50                 if(fork() == 0){
51                    aux3++;
52                 }
53                 // printf("%i:%i:%i:%i\n",index,aux1,aux2,aux3);
54              }else if (index == 4)
55              {
56                 if (aux1 == 0)
57                 {
58                    if (fork() == 0)
59                    {
60                       // printf("%i:%i:%i:%i\n",index,aux1,aux2,++aux3);
61                       int aux4 = 0;
62                       for (int i = 0; i < 2; ++i)
63                       {
64                          if(fork() == 0){
65                             aux4 += pow(2,i);
66                          }
67                       }
68                       // printf("%i:%i:%i:%i:%i\n",index,aux1,aux2,aux3,aux4);
69                    }else{
70                       // printf("%i:%i:%i:%i\n",index,aux1,aux2,aux3);
71                    }
72                 }else if (aux1 == 2)
73                 {
74                    for (int i = 0; i < 3; ++i)
75                    {
76                       if(fork() == 0){
77                          aux3 += pow(2,i);
78                       }
79                       if (aux3 > 4)
80                       {
81                          exit(0);
82                       }
83                    }
84                    // printf("%i:%i:%i:%i\n",index,aux1,aux2,aux3);
85                    int aux4 = 0;
86                    if (aux3 == 2)
87                    {
88                       if (fork() == 0)
89                       {
90                          aux4++;
91                       }
92                       // printf("%i:%i:%i:%i:%i\n",index,aux1,aux2,aux3,aux4);
93                    }else if (aux3 == 4)
94                    {
95                       if (fork() == 0)
96                       {
97                          exit(0);
98
99                       }else{
100                         // printf("%i:%i:%i:%i:%i\n",index,aux1,aux2,aux3,aux4);
101                         int aux5 = 0;
102                         for (int i = 0; i < 2; ++i)
103                         {
104                            if(fork() == 0){
105                               aux5 += pow(2,i);
106                            }
107                            if (aux5 > 2)
108                            {
109                               exit(0);
110                            }
111                         }
```

```
                        // printf("%i:%i:%i:%i:%i:%i\n",index,aux1,aux2,aux3,aux4,aux5);
                    }
                }
            }
        }
    }
}
else if (index == 2)
{
    if (fork() == 0)
    {
        // printf("%i:%i\n",index,++aux1);
        int aux2 = 0;
        if (fork() == 0)
        {
            // printf("%i:%i:%i\n",index,aux1,++aux2);
            int aux3 = 0;
            if (fork() == 0)
            {
                exit(0);
            }else{
                // printf("%i:%i:%i:%i\n",index,aux1,aux2,aux3);
                int aux4 = 0;
                if (fork() == 0)
                {
                    // printf("%i:%i:%i:%i:%i\n",index,aux1,aux2,aux3,++aux4);
                    int aux5 = 0;
                    if (fork() == 0)
                    {
                        aux5++;
                    }
                    // printf("%i:%i:%i:%i:%i:%i\n",index,aux1,aux2,aux3,aux4,aux5);
                }else{
                    // printf("%i:%i:%i:%i:%i\n",index,aux1,aux2,aux3,aux4);
                }
            }
        }else{
            // printf("%i:%i:%i\n",index,aux1,aux2);
            int aux3 = 0;
            if (fork() == 0)
            {
                exit(0);
            }else{
                // printf("%i:%i:%i:%i\n",index,aux1,aux2,aux3);
                int aux4 = 0;
                for (int i = 0; i < 2; ++i)
                {
                    if(fork() == 0){
                        aux4 += pow(2,i);
                    }
                    if (aux4 > 2)
                    {
                        exit(0);
                    }
                }
                // printf("%i:%i:%i:%i:%i\n",index,aux1,aux2,aux3,aux4);
                if (aux4 == 0 || aux4 == 2)
                {
                    int aux5 = 0;
                    if (fork() == 0)
                    {
                        aux5++;
                    }
                    // printf("%i:%i:%i:%i:%i:%i\n",index,aux1,aux2,aux3,aux4,aux5);
                }
            }
        }
    }else{
        // printf("%i:%i\n",index,aux1);
```

```
            }
        }
    }
    return 1;
}
```

6. Árbol E)

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

int main(void) {
    int id_proc;

    for (int i = 0; i < 2; i++)
    {
        if ((id_proc = fork()) == 0)
        {
            printf("Soy un proceso hijoA\n");
            if (i == 1)
            {
                for (int j = 0; j < 3; j++)
                {
                    if ((id_proc = fork()) == 0)
                    {
                        printf("Soy un proceso hijoB\n");
                        if (j == 0)
                        {
                            for (int k = 0; k < 3; k++)
                            {
                                if ((id_proc = fork()) == 0)
                                {
                                    printf("Soy un proceso hijoC\n");
                                    if (k == 0)
                                    {
                                        for (int l = 0; l < 2; l++)
                                        {
                                            if ((id_proc = fork()) == 0)
                                            {
                                                printf("Soy un proceso hijoD\n");
                                                if (l == 0)
                                                {
                                                    if ((id_proc = fork()) == 0)
                                                    {
                                                        printf("Soy un proceso hijoE\n");
                                                        for (int m = 0; m < 2; m++)
                                                        {
                                                            if ((id_proc = fork()) == 0)
                                                            {
                                                                printf("Soy un proceso hijoF\n");
                                                                if (m == 0)
                                                                {
                                                                    for (int n = 0; n < 3; n++)
                                                                    {
                                                                        if ((id_proc = fork()) == 0)
                                                                        {
                                                                            printf("Soy un proceso hijoG\n");
                                                                            exit(0);
                                                                        }
                                                                    }
                                                                }
                                                                else if (m == 1)
                                                                {
                                                                    exit(0);
```

```
                                    }
                                    exit(0);
                                }
                            }
                            exit(0);
                        }
                    }
                    else if (l == 1)
                    {
                        if ((id_proc = fork()) == 0)
                        {
                            printf("Soy un proceso hijoH\n");
                            for (int m = 0; m < 3; m++)
                            {
                                if ((id_proc = fork()) == 0)
                                {
                                    printf("Soy un proceso hijoI\n");
                                    if (m == 2)
                                    {
                                        for (int n = 0; n < 3; n++)
                                        {
                                            if ((id_proc = fork()) == 0)
                                            {
                                                printf("Soy un proceso hijoJ\n");
                                                exit(0);
                                            }
                                        }
                                    }
                                    else
                                    {
                                        exit(0);
                                    }
                                    exit(0);
                                }
                            }
                            exit(0);
                        }
                    }
                    exit(0);
                }
            }
        }
        else if (k == 1)
        {
            if ((id_proc = fork()) == 0)
            {
                printf("Soy un proceso hijoK\n");
                if ((id_proc = fork()) == 0)
                {
                    printf("Soy un proceso hijoL\n");
                    for (int l = 0; l < 4; l++)
                    {
                        if ((id_proc = fork()) == 0)
                        {
                            printf("Soy un proceso hijoM\n");
                            if (l == 3)
                            {
                                for (int m = 0; m < 4; m++)
                                {
                                    if ((id_proc = fork()) == 0)
                                    {
                                        printf("Soy un proceso hijoN\n");
                                        exit(0);
                                    }
                                }
                            }
                            else
                                exit(0);
                            exit(0);
```

```
                        }
                    }
                    exit(0);
                }
                else
                    exit(0);
                exit(0);
            }
            else
                exit(0);
        }
        else if (k == 2)
        {
            if ((id_proc = fork()) == 0)
            {
                printf("Soy un proceso hijoO\n");
                exit(0);
            }
        }
        exit(0);
    }
  }
}
else if (j == 1)
{
  for (int k = 0; k < 2; k++)
  {
    if ((id_proc = fork()) == 0)
    {
      printf("Soy un proceso hijoP\n");
      exit(0);
    }
  }
}
else if (j == 2)
{
  if ((id_proc = fork()) == 0)
  {
    printf("Soy un proceso hijoQ\n");
    for (int k = 0; k < 2; k++)
    {
      if ((id_proc = fork()) == 0)
      {
        if (k == 0)
        {
          for (int l = 0; l < 2; l++)
          {
            if ((id_proc = fork()) == 0)
            {
              printf("Soy un proceso hijoR\n");
              if (l == 0)
              {
                for (int m = 0; m < 2; m++)
                {
                  if ((id_proc = fork()) == 0)
                  {
                    printf("Soy un proceso hijoS\n");
                    exit(0);
                  }
                }
              }
              else if (l == 1)
              {
                for (int m = 0; m < 2; m++)
                {
                  if ((id_proc = fork()) == 0)
                  {
                    printf("Soy un proceso hijoT\n");
                    if (m == 0)
```

```
                                    exit(0);
                                else if (m == 1)
                                {
                                    for (int n = 0; n < 2; n++)
                                    {
                                        if ((id_proc = fork()) == 0)
                                        {
                                            printf("Soy un proceso hijoU\n");
                                            exit(0);
                                        }
                                    }
                                }
                                exit(0);
                            }
                        }
                    }
                    exit(0);
                }
            }
        }
        else if (k == 1)
        {
            for (int l = 0; l < 3; l++)
            {
                if ((id_proc = fork()) == 0)
                {
                    printf("Soy un proceso hijoV\n");
                    if (l == 0)
                    {
                        exit(0);
                    }
                    else if (l == 1)
                    {
                        for (int m = 0; m < 3; m++)
                        {
                            if ((id_proc = fork()) == 0)
                            {
                                printf("Soy un proceso hijoW\n");
                                if (m == 0)
                                {
                                    exit(0);
                                }
                                else if (m == 1)
                                {
                                    for (int n = 0; n < 3; n++)
                                    {
                                        if ((id_proc = fork()) == 0)
                                        {
                                            printf("Soy un proceso hijoX\n");
                                            exit(0);
                                        }
                                    }
                                }
                                else if (m == 2)
                                {
                                    exit(0);
                                }
                                exit(0);
                            }
                        }
                    }
                    else if (l == 2)
                    {
                        for (int m = 0; m < 2; m++)
                        {
                            if ((id_proc = fork()) == 0)
                            {
                                printf("Soy un proceso hijoY\n");
                                if (m == 0)
```

```
265                                                                 {
266                                                                    exit(0);
267                                                                 }
268                                                                 else if (m == 1)
269                                                                 {
270                                                                    for (int n = 0; n < 2; n++)
271                                                                    {
272                                                                       if ((id_proc = fork()) == 0)
273                                                                       {
274                                                                          printf("Soy un proceso hijoZ\n");
275                                                                          exit(0);
276                                                                       }
277                                                                    }
278                                                                 }
279                                                                 exit(0);
280                                                              }
281                                                           }
282                                                        }
283                                                        exit(0);
284                                                     }
285                                                  }
286                                               }
287                                               exit(0);
288                                            }
289                                         }
290                                         exit(0);
291                                      }
292                                   }
293                                   exit(0);
294                                }
295                             }
296                          }
297                       }
298                       else
299                          exit(0);
300                 }
301                 exit(0);
302              }
```