Program 1

#include <sys/wait.h>

#include <stdio.h>

#include <unistd.h>

*int* value = 5;

*int* main()

{

*pid\_t* pid;

    pid = fork();

    if (pid == 0)

    { */\* child process \*/*

        value += 15;

        return 0;

    }

    else if (pid > 0)

    { */\* parent process \*/*

        wait(NULL);

        printf("PARENT: value = %d", value); */\* LINE A \*/*

        printf("\n");

        return 0;

    }

}

OUTPUT

Text

Description automatically generated with low confidence

Program 2

#include <sys/wait.h>

#include <stdio.h>

#include <unistd.h>

*int* main()

{

*int* i;

    fork();

    fork();

    fork();

    printf("pid %d \n", getpid());

    return 0;

}

OUTPUT

Text

Description automatically generated

Program 3

#include <sys/wait.h>

#include <stdio.h>

#include <unistd.h>

*int* main()

{

*pid\_t* pid;

*/\* fork a child process \*/*

    pid = fork();

    if (pid < 0)

    { */\* error occurred \*/*

        fprintf(stderr, "Fork Failed");

        return 1;

    }

    else

    { */\* parent process \*/*

*/\* parent will wait for the child to complete \*/*

        wait(NULL);

        printf("Child Complete");

    }

    return 0;

}

OUTPUT

Text

Description automatically generated with medium confidence

Program 4

#include <sys/wait.h>

#include <stdio.h>

#include <unistd.h>

*int* main()

{

*pid\_t* pid, pid1;

*/\* fork a child process \*/*

    pid = fork();

    if (pid < 0)

    { */\* error occurred \*/*

        fprintf(stderr, "Fork Failed");

        return 1;

    }

    else if (pid == 0)

    { */\* child process \*/*

        pid1 = getpid();

        printf("child: pid = %d\n", pid); */\* A \*/*

        printf("child: pid1 = %d\n", pid1); */\* B \*/*

    }

    else

    { */\* parent process \*/*

        pid1 = getpid();

        printf("parent: pid = %d\n", pid); */\* C \*/*

        printf("parent: pid1 = %d\n", pid1); */\* D \*/*

        wait(NULL);

    }

    return 0;

}

OUTPUT

Text

Description automatically generated

Program 5

#include <sys/wait.h>

#include <stdio.h>

#include <unistd.h>

#define SIZE 5

*int* nums[SIZE] = {0, 1, 2, 3, 4};

*int* main()

{

*int* i;

*pid\_t* pid;

    pid = fork();

    if (pid == 0)

    {

        for (i = 0; i < SIZE; i++)

        {

            nums[i] \*= -i;

            printf("CHILD: %d \n", nums[i]); */\* LINE X \*/*

        }

    }

    else if (pid > 0)

    {

        wait(NULL);

        for (i = 0; i < SIZE; i++)

            printf("PARENT: %d \n", nums[i]); */\* LINE Y \*/*

    }

    return 0;

}

OUTPUT

Text

Description automatically generated